Implicit Imitation in Multiagent Reinforcement Learning

Bob Price and Craig Boutilier
ICML-99

Slides: Dana Dahlstrom
CSE 254, UCSD
2002.04.23

Overview

- Learning by imitation entails watching a mentor perform a task.
- The approach here combines direct experience with an environment model extracted from observations of a mentor.
- This approach shows improved performance and convergence compared to a non-imitative reinforcement learning agent.
Background

- Other multi-agent learning schemes include:
  - explicit teaching (demonstration)
  - sharing of privileged information
  - elaborate psychological imitation theory
- All these require explicit communication, and usually voluntary cooperation by the mentor.
- A common thread: the observer explores, guided by the mentor.

Implicit Imitation

In *implicit* imitation, the learner observes the mentor’s state transitions but not its actions.

- No demands are made of the mentor beyond ordinary behavior.
  - no voluntary cooperation
  - no explicit communication
- The learner can take advantage of multiple mentors.
- The learner is not forced to follow in the mentor’s footsteps.
  - can learn from negative examples without paying a penalty
Markov Decision Processes

A preliminary assumption: the learner and mentor(s) act concurrently in a single environment, but their actions are noninteracting. Therefore the underlying multi-agent Markov decision process (MMDP) can be factored into separate single-agent MDPs \((S, A, Pr, R)\).

- \(S\) is the set of states.
- \(A\) is the set of actions.
- \(Pr(t|s,a)\) is the probability of transitioning to state \(t\) when performing action \(a\) in state \(s\).
- \(R(s,a,t)\) is the reward received when action \(a\) is performed in state \(s\) and there is a transition to state \(t\).

Further Assumptions

- The learner and mentor have identical state spaces: \(S = S_m\)
- All the mentor’s actions are available to the learner: \(A \supseteq A_m\)
- The mentor’s transition probabilities apply to the learner: for all states \(s\) and \(t\), if \(a \in A_m\) then \(Pr(t|s,a) = Pr_m(t|s,a)\).
- The learner knows its own reward function \(R(s,a,t) = R(s)\).
- The learner can observe the mentor’s state transitions \((s,t)\).
- The horizon is infinite with discount factor \(\gamma\).
The Reinforcement Learning Task

The task is to find a policy \( \pi : S \to A \) that maximizes the total discounted reward. Under such an optimal policy \( \pi^* \), the total discounted reward \( V^*(s) \) at state \( s \) is given by the Bellman equation:

\[
V^*(s) = R(s) + \gamma \max_{a \in A} \left\{ \sum_{t \in S} \Pr(t|s, a) V^*(t) \right\}
\]  

(1)

- Given samples \( (s, a, t) \) the agent could
  - estimate an action-value function directly via Q-learning, or
  - estimate \( \Pr \) and solve for \( V^* \) in Equation (1).
- Prioritized sweeping converges on a solution to the Bellman equation as its estimate of \( \Pr \) improves.

Estimating the Transition Probabilities

The transition probabilities can be estimated by observed frequencies

\[
\hat{\Pr}(t|s, a) = \frac{\text{count}\left( (s, a, t) \right)}{\sum_{t' \in S} \text{count}\left( (s, a, t') \right)}
\]

For all states \( t \), as the number of times the learner has performed action \( a \) in state \( s \) approaches infinity, the estimate \( \hat{\Pr}(t|s, a) \) converges to the actual probability \( \Pr_m(t|s, a) \).
Estimating the Mentor’s Transition Probabilities

Assuming the mentor uses a stationary, deterministic policy $\pi_m$,

$$\Pr_m(t|s) = \Pr_m(t|s, \pi_m(s))$$

In this case the mentor’s transition probabilities too can be estimated by observed frequencies

$$\hat{\Pr}_m(t|s) = \frac{\text{count}_m((s, t))}{\sum_{t' \in S} \text{count}_m((s, t'))}$$

For all states $t$, as the mentor’s visits to state $s$ approach infinity, the estimate $\hat{\Pr}_m(t|s)$ converges to the actual probability $\Pr_m(t|s)$.

Augmenting the Bellman Equation

**Lemma:** The imitation learner’s state-value function is specified by the augmented Bellman equation

$$V^*(s) = R(s) + \gamma \max \left\{ \sum_{t \in S} \Pr_m(t|s)V^*(t), \max_{a \in A} \left\{ \sum_{t \in S} \Pr(t|s, a)V^*(t) \right\} \right\} \quad (2)$$

**Proof idea:** Since $\Pr_m(t|s) = \Pr(t|s, \pi_m(s))$, the first summation is equal to the second when $a = \pi_m(s)$. We know $\pi_m(s) \in A$ because $\pi_m(s) \in A_m$ and $A_m \subseteq A$; therefore the first summation is redundant and Equation 2 simplifies to Equation 1.

Extension to multiple mentors is straightforward.
Augmented Bellman Backups

Bellman backups update state-value estimations. The augmented Bellman equation suggests the update rule

\[
\hat{V}(s) \leftarrow (1 - \alpha)\hat{V}(s) + \alpha R(s) + \alpha \gamma \max \left\{ \sum_{t \in S} \hat{\Pr}_m(t|s)\hat{V}(t), \max_{a \in A} \left\{ \sum_{t \in S} \hat{\Pr}(t|s,a)\hat{V}(t) \right\} \right\}
\]

where \(\alpha\) is the learning rate.

Confidence Estimation

The learner must rely on estimates \(\hat{\Pr}(t|s,a)\) and \(\hat{\Pr}_m(t|s)\). It is best to account for the unreliability of these estimates.

- \(\hat{\Pr}(t|s,a)\) and \(\hat{\Pr}_m(t|s)\) are multinomial distributions; assume Dirichlet priors over them.

- Compute the learner’s value function \(V(s)\) and the mentor’s value function \(V_m(s)\) within suitable confidence intervals; let \(v^-\) and \(v_m^-\) be the lower bounds of these intervals.

- If \(v_m^- < v^-\), then ignore mentor observations; either the mentor’s policy is suboptimal or confidence in \(\hat{\Pr}_m\) is too low.
Accommodating Action Costs

When the reward function $R(s, a)$ depends on the action, how can it be applied to mentor observations without knowing the mentor’s action?

Let $\kappa(s)$ denote an action whose transition distribution at state $s$ has minimum Kullback-Leibler (KL) distance from $Pr_m(t|s)$:

$$\kappa(s) = \arg\min_{a \in A} \left\{ -\sum_{t \in S} Pr(t|s, a) \log Pr_m(t|s) \right\} \quad (3)$$

Using the guessed mentor action $\kappa(s)$, the augmented Bellman equation can be rewritten as

$$V^*(s) = \max \left\{ \begin{array}{l} R(s, \kappa(s)) + \gamma \sum_{t \in S} Pr_m(t|s) V^*(t), \\
R(s, a) + \gamma \max_{a \in A} \left\{ \sum_{t \in S} Pr(t|s, a) V^*(t) \right\} \end{array} \right\}$$

Prioritized Sweeping

In prioritized sweeping (Moore & Atkeson, 1993) $N$ backups are performed per transition.

- Maintain a queue of states whose value would change upon backup, prioritized by the magnitude of change.

- At each transition $\langle s, t \rangle$:
  1. If a backup would change its value more than a threshold amount $\theta$, insert $s$ into the queue.
  2. Do backups for the top $N$ states in the queue, inserting their graphwise predecessors (or updating their priorities) if backups would change their values more than $\theta$. 

Implicit Imitation in Prioritized Sweeping

To incorporate implicit imitation into prioritized sweeping:

- do backups for mentor transitions as well as learner transitions
- use augmented Bellman instead of standard Bellman backups
- ignore the mentor-derived model when confidence in it is too low

Implicit Imitation in Q-Learning

Model extraction can be incorporated into algorithms other than prioritized sweeping, such as Q-learning.

- Augment the action space with a placeholder action $a_m \in A$.
- For each transition $(s, t)$ use the update rule:

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha \left( R(t) + \gamma \max_{a' \in A} Q(t, a') \right)$$

where $a = a_m$ for observed mentor transitions, and $a$ is the action performed by the learner otherwise.
Action Selection
An $\varepsilon$-greedy action selection policy ensures exploration:
- with probability $\varepsilon$, pick an action uniformly at random
- with probability $1 - \varepsilon$, pick the greedy action

The “greedy action” is here defined as the $a$ whose estimated distribution $\hat{\Pr}(t|s, a)$ has minimum KL distance from $\hat{\Pr}_m(t|s)$.

Experimental Setup
To evaluate their technique, the authors simulated three different agents:
- an expert mentor following an $\varepsilon$-greedy policy with $\varepsilon \in \Theta(0.01)$
- an imitative prioritized sweeping learner observing the mentor
- a non-imitative prioritized sweeping learner

They compare the imitation learner’s performance to that of the non-imitation learner, as a control.
- The learners use the same parameters, including a fixed number of backups per sample.
- The learners’ $\varepsilon$ decays over time.
Figure 1: Performance in a $10 \times 10$ grid world with 10% noisy actions.

Figure 2: Imitation vs. control for different grid-world parameters.
Figure 5: A “complex maze” grid world.

Figure 6: Performance in the grid world of Figure 5.
Figure 7: A “perilous shortcut” grid world.

Figure 8: Performance in the grid world of Figure 7.
Figure 9: A grid world with multiple mentors whose trajectories are different from, but overlapping with, the learner’s target trajectory.

Figure 10: Performance in the grid world of Figure 9.
Summary: Assumptions

- Multiple agents’ actions are noninteracting.
- The learner and mentor have “similar” capabilities:
  - Their state spaces are identical.
  - All actions the mentor can take are available to the learner.
  - All the mentor’s transition probabilities apply to the learner.
- The learner knows its own reward function.
- The learner can observe the mentor’s state transitions.
  - For convergence, the observation period is indefinite.
  - The mentor’s policy is stationary.

Summary: Results

Implicit imitation shows:

- improvement over standard learning (given an expert mentor)
- tolerance to noise (Figures 1 and 2)
- the ability to integrate subskills from multiple mentors (Figure 10)
- benefits that increase with problem difficulty (Figures 5 and 6)