Active Imitation Learning

Aaron P. Shon and **Deepak Verma** and **Rajesh P. N. Rao**
Department of Computer Science and Engineering
University of Washington Seattle, WA 98195
{*aaron,deepak,rao*}*@cs.washington.edu*

Abstract

Imitation learning, also called *learning by watching* or *programming by demonstration*, has emerged as a means of accelerating many reinforcement learning tasks. Previous work has shown the value of imitation in domains where a single mentor demonstrates execution of a known optimal policy for the benefit of a learning agent. We consider the more general scenario of learning from mentors who are themselves agents seeking to maximize their own rewards. We propose a new algorithm based on the concept of transferable utility for ensuring that an observer agent can learn efficiently in the context of a selfish, not necessarily helpful, mentor. We also address the questions of when an imitative agent should request help from a mentor, and when the mentor can be expected to acknowledge a request for help. In analogy with other types of active learning, we call the proposed approach *active imitation learning*.

Introduction

Despite its power, a key limitation of traditional reinforcement learning is its assumption that a learning agent's only source of information is a reward signal from the environment. Imitation learning is attractive for applications where an agent can expect to find skilled teachers willing to demonstrate good policies through a state space. Recent efforts have shown how imitation speeds up reinforcement learning in simulated maze environments (Price 2003), driving tasks (Abbeel & Ng 2004), and in real-world robotic control (Schaal 1997).

But imitation is not a panacea. Observers cannot always assume that mentors are intrinsically helpful—presumably, mentors are selfish agents wishing to maximize their own cumulative reward from the environment. The observer may need to perform some intervention to request help from the mentor at particularly opportune times, offering the mentor something in exchange for the mentor demonstrating a sample of environmental dynamics. This motivates our development of *active imitation learning*. In analogy with other paradigms for active learning, active imitation learning asks:

1. **When** should a naive agent ask for help from a mentor?
2. **How much** should the naive agent be willing to pay in exchange for information?
3. **When** should a mentor respond to a request for help?
4. **Whom** should the agent ask, given a set of possible mentors?

In short: *How can an imitative agent select actions that maximize the combined value of information gained from its own exploration and from the behaviors of other selfish agents?*

Potential application domains include (Fig. 1(a)):

- *Robotics:* For example, one robot might show another how to find its way home, in exchange for being shown the way to the nearest recharging station.
- *Online agents:* An agent that prices items online might ask another where to find cheap airline tickets, in exchange for a percentage of the revenue.
- *Human-computer interaction:* A poker "bot" might induce an expert human player to show it when to bet and when to fold via monetary reward; a stock-selecting agent might induce human managers to show good stock picking strategies, conditioned on a cut of the profits.

Model parameters and assumptions

Let $\mathcal{S}, \mathcal{A}$ be single-valued, discrete state and action spaces. Let $R : s \in \mathcal{S} \mapsto r \in \mathbb{R}$ represent a reward function mapping states to scalar rewards. The function τ parameterizes the environmental dynamics $P\left(s_{t+1}|s_t, a_t\right)$ giving the probability of moving from state s_t to s_{t+1} when action a_t is applied at time t. The observer and mentor are thus represented as Markov decision processes (MDPs) $\langle \mathcal{S}, \mathcal{A}, R, \tau \rangle$. The goal of learning in an MDP (with a finite horizon) is to learn a policy $\pi : s \in \mathcal{S} \mapsto a \in \mathcal{A}$ that maximizes cumulative discounted reward starting from state s_0:

$$V_\pi(s_0) = \sum_{t=1}^{T} \gamma^t R(s_t) \tag{1}$$

Let $\mathcal{H} = \{H^1 \ldots H^n\}$ represent "state histories" the agent has observed of moving around in the environment; these histories can be derived from the agent's own interactions with the environment, or, in the case of imitation, from observations of mentor agent(s).

We adopt the "implicit imitation" framework of (Price 2003; Price & Boutilier 1999). Communication between

agents is only possible through the environment. We assume the mentor cannot directly relay its policy to the observer agent – for example, a human mentor cannot convey his or her policy for returning a tennis serve to a robot. The observer attempts to improve its estimate of the environmental parameters $\hat{\tau}$ (and thus improve its policy) by watching the behavior of the mentor. As in (Price 2003; Price & Boutilier 1999), we assume:

1. The mentor is fully observable.
2. The reward function is completely known to all agents and identical for all agents.
3. The observer has access to its own state-action history $s^o_{1:t}, a^o_{1:t-1}$. and the state history (but not actions) of the mentor $s^m_{1:t}$.
4. The action spaces of mentor and observer are equivalent (or correspondences can be made between them): $\mathcal{A}^o \equiv \mathcal{A}^m$
5. The above imply $\pi^*_m = \pi^*_o$—the optimal policy for the mentor is identical to the optimal policy for the observer.

Our point of departure from Price and Boutilier's original framework is to break the assumption of a helpful mentor, in the sense that the mentor must be induced to show interesting samples from the environment.

We analyze mentor and observer behavior in terms of *iterative games*. Each agent playing in the game follows a *trajectory* from a starting state at time 0, s_0, to an end state, or goal, s_T at time T. T may vary across individual games and across individual agents. An agent's state history for game h, $H^h = \{s_0(h) \ldots s_T(h)\}$, resulting from playing according to policy π, determines the value received by playing game h: $V^h_\pi(s_0)$. Learning in our model occurs using the augmented backup equation in (Price & Boutilier 1999), where an additional $\max$ operation backs up the value of states visited by the mentor:

$$V(s) = R(s) + \gamma \max \left\{ \max_{a \in \mathcal{A}} \left(\sum_{s' \in \mathcal{S}} P_o(s'|s,a) V(s') \right), \sum_{s' \in S} P_m(s'|s) V(s') \right\} \quad (2)$$

where $P_o(s'|s,a) = \tau$ and $P_m(s'|s)$ represents the Markov chain induced by the mentor as it moves through the environment. Actions a generated by the mentor policy π_m are estimated as $a = \text{argmin}_{\mathcal{A}} \text{KL}(P(s^m_{t+1}|s^m_t)||P(s^o_{t+1}|a, s^o_t))$, the action that minimizes the KL-divergence between the mentor's distribution $P(s^m_{t+1}|s^m_t)$ and the observer's distribution $P(s^o_{t+1}|a, s^o_t)$ if the mentor policy has higher expected value at state s.

Inducing demonstrations with side payments

The game-theoretic notion of *side payments* is often used in bargaining problems between selfish agents. We show that side payments promote learning in imitation learning agents. We assume that 1) the mentor can observe the observer's state and 2) the observer can make a real-valued side payment to the mentor. Side payments need not involve physical exchange; in many real-world cases, side payments are informational in nature (agreeing to swap a future demonstration of a skill for a demonstration of another skill in the present), or hardwired rewards built in by evolution (as when an infant's smile elicits attention and demonstration from an adult caregiver).

The value of demonstration

To properly value side payments, we must define the value of demonstration. One thread of reinforcement learning research has considered the *value of perfect information* (VOI) for a single autonomous agent exploring an environment, attempting to resolve the well-known exploration-exploitation tradeoff (Dearden, Friedman, & Russell 1998; Dearden, Friedman, & Andre 1999). As in these earlier papers, we employ concepts from information value theory (Howard 1966) to determine the value of observing a sample history determined by the mentor's policy; that is, we propose computing the *value of demonstration*, or VOD. For the remainder of the paper, we write from the observer's perspective; e.g., we write "our estimate" as shorthand for "the observer's estimate." Because knowledge of environmental parameters is imperfect, our analysis is based on the observer's current estimate of the value function $\hat{V}_o(s)$. Initially, $\hat{V}_o(s)$ is a poor estimate of the true value function given by executing the observer's policy π_o at state s: $V_{\pi_o}(s)$. For a Bayesian estimate of $\hat{V}_o(s)$, we maintain a set of *hyperparameters* θ associated with the value function at s. We begin by quantifying the expectation wrt θ of $\Delta\hat{V}_o(s)$, the change in value of the observer's policy assuming that our current estimate of the value of the mentor's policy, $\hat{V}_m(s)$, is correct:

$$\begin{aligned} \text{E}_\theta \left[\Delta \hat{V}_o(s) \right] &= \sum_{k=1}^{\infty} \text{E}_\theta \left[\hat{V}_m(s) - \hat{V}_o(s) \right] \delta^{k-1} \quad (3) \\ &= \frac{1}{1-\delta} \left(\text{E}_{\mu_m, \sigma_m} \left[\hat{V}_m(s) \right] - \text{E}_{\text{Dir}_o} \left[\hat{V}_o(s) \right] \right) \end{aligned}$$

$\text{E}_{\text{Dir}_o} \left[\hat{V}_o(s) \right]$ is the expected value of s given π_o, determined by sampling MDPs using the set of Dirichlet counts for $\hat{\tau}$ (Dearden, Friedman, & Andre 1999). $0 \leq \delta < 1$ is a discount factor over games (similar to γ as a within-game discount). By the definition of expectation and making the simplifying approximation that each observed mentor state history is conditionally independent given the mentor's value function (and hence the mentor's policy), we have:

$$\begin{aligned} \text{E}_{\mu_m, \sigma_m} \left[\hat{V}_m(s) \right] &= \int_\theta P \left(\hat{V}_m(s) | H^1_m \ldots H^n_m \right) \hat{V}_m(s) d\theta \\ &= \int_\theta P \left(H^1_m \ldots H^n_m | \hat{V}_m(s) \right) P \left(\hat{V}_m(s) \right) \hat{V}_m(s) d\theta \\ &= \int_\theta \prod_{h=1}^{n} P \left(H^h_m | \hat{V}_m(s) \right) P \left(\hat{V}_m(s) \right) \hat{V}_m(s) d\theta \quad (4) \end{aligned}$$

where $\mu_m, \sigma_m \in \theta$ are the maximum likelihood estimates of the mean and standard deviation of the mentor's value function at s.

The estimate in (3) is one candidate for computing VOD. We call this estimate *policy regret*. One critical question is how to represent and compute $P\left(\hat{V}_m(s)\right)$ and $P\left(H_m^h|\hat{V}_m(s)\right)$. Following the justifications in (Dearden, Friedman, & Russell 1998; Mannor *et al.* 2004), we model the value function conditioned on the history using a normal distribution [1]. For the estimate in (3), we need only model the mean and standard deviation μ_m, σ_m of the mentor's value function. By contrast, the Bayesian algorithms we present below use uncertainty in the parameters μ_m, σ_m to compute VOD.

Bayesian modeling has the advantage that, by sampling from the priors on μ_m, σ_m, we can compute $\mathrm{Var}\left(\mathrm{E}_\theta\left[\hat{V}_m(s)\right]\right)$, the variance in the mean estimator for value of π_m. Inspired by (Dearden, Friedman, & Russell 1998), we define the function $\mathrm{Gain}(\mu(s), \hat{\mu}(s)) = |\mu(s) - \hat{\mu}(s)|$, which represents the value of discovering that the expected value at state s is actually $\mu(s)$ rather than the current estimate $\hat{\mu}(s)$. The conjugate prior distribution for the normal is the so-called "normal-gamma" distribution, whose parameters $\{\alpha, \beta, \mu_0\}$ are included in θ. For $\mu_m(s), \eta_m(s)$, $P\left(\hat{V}_m(s)\right)$ (with $\eta_m(s) = \frac{1}{\sigma_m(s)}$ the precision of the distribution over $\hat{V}_m(s)$) we have:

$$\hat{V}_m(s) \sim \mathcal{N}\left(\mu_m(s), 1/\eta_m(s)\right) \quad (5)$$

$$P\left(\mu_m(s), \eta_m(s)\right) \propto \eta_m^{\frac{1}{2}} \exp\left[-\frac{1}{2}\lambda\eta_m\left(\mu_m - \mu_0\right)^2\right] \quad (6)$$
$$\eta_m^{\alpha-1} \exp\left[-\beta\eta_m\right]$$

We can compute the expected gain of seeing another example trajectory from the mentor by sampling N possible means $\mu_i, i \in 1 \ldots N$, for the mentor's value function given the hyperparameters at s, and computing how much value we would gather over an infinite series of games with discount factor δ given the expectation of the samples:

$$\mu_i(s) \sim \mathcal{N}(\mu_0(s), \eta_i(s))$$
$$\mathrm{VOD}(s) \approx \sum_{k=1}^{\infty}\left[\frac{1}{N}\sum_{i=1}^{N}\mathrm{Gain}(\mu_i(s), \mu_0(s))\right]\delta^{k-1} \quad (7)$$

As the number of demonstrations from state s grows, Eqn. 7 goes to 0 (Fig. 2(d)). We call the estimate in Eqn. 7 *Bayesian VOD*. Given a set of n observed mentor state histories, $H^1 \ldots H^n$, we can update the normal-gamma hyperparameters θ to a new set θ' using a well-known result (DeGroot 1986):

$$\bar{V}_m = \frac{1}{n}\sum_{h=1}^{n}\sum_{t=1}^{T}\gamma^t R\left(s_t^h\right) \quad (8)$$

$$\tilde{V}_m = \frac{1}{n}\sum_{h=1}^{n}\left(\sum_{t=1}^{T}\gamma^t R\left(s_t^h\right)\right)^2 \quad (9)$$

$$\mu_0' = \frac{\lambda\mu_0 + n\bar{V}_m}{\lambda + n} \quad (10)$$

$$\lambda' = \lambda + n \quad (11)$$

$$\alpha' = \alpha + \frac{1}{2}n \quad (12)$$

$$\beta' = \beta + \frac{1}{2}n\left(\tilde{V}_m - \bar{V}_m^2\right) + \frac{n\lambda\left(\bar{V}_m - \mu_0\right)^2}{2(\lambda + n)} \quad (13)$$

The question of when to readjust the hyperparameters requires a decision about how reliable individual observations of mentor state sequences are, and how much processing power the observer can devote. We currently readjust hyperparameters after every new mentor history is added. In principle, a hierarchical model (including uncertainty in the hyperparameters) could determine optimal times to recompute hyperparameters.

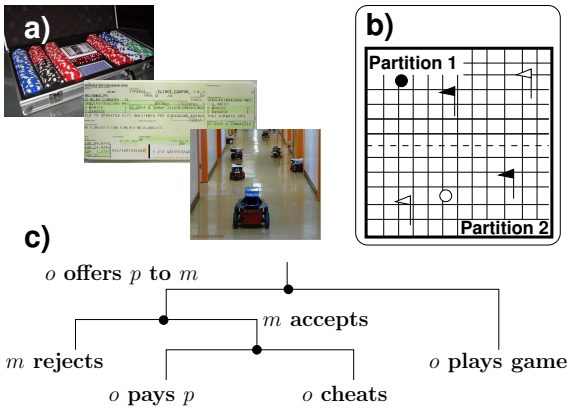

Figure 1: **Imitation and active learning:** (a) Potential domains where active imitation might be useful include online games where agents can ask for demonstrations, online pricing of commodities like airline tickets where agents can offer others incentives to show how to find low-price fares, and robot navigation where agents can ask others for help. (b) In a multiagent environment where an agent can be both mentor and observer, local knowledge of partitions on the state space leads to a Prisoner's Dilemma situation. We propose use of side payments to achieve optimality in such cases. (c) Overview of decisions made in active imitation learning. See text.

Relation to the Prisoner's Dilemma

Consider the problem of two agents trying to recover flags in the grid-world environment shown in Fig. 1(b). The black agent ("B") tries to recover two black flags, and the white agent ("W") tries to recover two white flags. The transition dynamics of the world are broken into two different partitions. Black is an expert in one partition, and White is an ex-

[1] This is justified using the central limit theorem, assuming that the number of visited states is large and that reward values are distributed iid (approximately).

pert in another. One white flag and one black flag are placed in each partition. A small cost is associated with each step an agent takes in the world until it has both flags (at which point it receives a reward).

However, each agent can choose to watch what the other agent does, informing it about likely state sequences through the environment (and thereby, as shown in (Price 2003; Price & Boutilier 1999), giving information about environmental dynamics). Demonstrating enough samples through the environment for an observing agent to learn incurs a cost for the demonstrating mentor agent, say a cost of 1. If an agent collects both of its flags, it receives a reward of 2. For a single iteration of such a game, this leads to the following payoff matrix:

(W payoff, B payoff)	W demonstrates	W watches
B demonstrates	(1,1)	(2,-1)
B watches	(-1,2)	(0,0)

This is clearly a Prisoner's Dilemma game (Dresher 1961). The hallmark of Prisoner's Dilemma problems is that their Nash equilibrium strategies are not Pareto optimal. That is, if each agent plays optimally according to a minimax criterion (the strategy pair ⟨ W watches, B watches ⟩), overall utility in the system is not maximized (the strategy ⟨ W demonstrates, B demonstrates ⟩ is never played). We argue that side payments between agents can achieve Pareto optimality in imitation games.

In single-shot imitative games of this form, as in the Prisoner's Dilemma, it makes no sense to demonstrate to the other player. For iterative games, however, demonstrative exchanges between agents can lead to higher combined payoff. We do not discuss trust issues, except to note that very simple, deterministic strategies (such as refusing to interact in the future) discourage cheating on iterative Prisoner's Dilemma problems (Axelrod 2006). Fig. 1(c) depicts the strategies open to each agent in active imitation games: the observer can decide whether or not to propose a side payment to the mentor; the mentor can accept or reject the offer; and the observer can honor the agreement or cheat.

Offering and accepting side payments

Algorithm 1 computes an optimal side payment for the observer. At the beginning of a game, the observer can decide whether or not to offer a side payment to the mentor. If the observer determines that a side payment is warranted (Fig. 4), it computes a side payment p to be offered to the mentor for moving from its current state s_m to the observer's state s, then continuing to execute according to $\pi_m(s)$. Side payments are computed according to the Bayesian VOD as defined in (7). The observer maintains parameters $\mu_{\text{pay}}(s, s_m), \sigma_{\text{pay}}(s, s_m)$ representing the payoff amounts accepted in the past by the mentor, given acceptance of the offer, conditioned on s, s_m. Another μ, σ can be maintained for s, s_m conditioned on rejection of the offer. A simple method that works in practice is to assume that the mentor's acceptance decision is deterministic, and use binary search to determine the minimum payment at which the mentor will accept. Then the observer should offer $\min\left(VOD(s), \mathrm{E}_{\mu_{\text{pay}}(s,s_m),\sigma_{\text{pay}}(s,s_m)}\left[\text{pay}|\text{accept}\right]\right)$.

Algorithm 1 Observer computes side payment p

Input: A set of hyperparameters $\theta = \{\alpha, \beta, \lambda, \mu_0\}$, number of samples N, observer state s, and mentor state s_m.

for $i = 1$ to N **do**
  $\eta_i \sim \text{Gamma}(\alpha, \beta, \lambda)$ /* Generate N samples */
  $\mu_i \sim \mathcal{N}(\mu_0, 1/\eta_i)$
end for
$v \leftarrow \frac{1}{(1-\delta)N} \sum_{i=1}^{N} |\hat{\mu}_i - \mu_0|$ /* Compute $VOD(s)$ */
$p \leftarrow \min\left(v, \mathrm{E}_{\mu_{\text{pay}}(s,s_m),\sigma_{\text{pay}}(s,s_m)}\left[\text{pay}|\text{accept}\right]\right)$

Algorithm 2 determines whether a (selfish) mentor should accept, in the sense that a mentor not responding in this way to a proposed side payment will always achieve lower cumulative reward than a mentor that does employ the algorithm. Given a proposed payment p to move to state s from s_m, the mentor adds p to its reward at s to compute a new policy π'_m. Then the mentor simulates running $\pi'_m(s_m)$ using a number of samples N. If some fraction ϵ of the N samples reach s, the mentor has effectively decided to accept.

Algorithm 2 Mentor computes acceptance

Input: A proposed payoff p at state s, a transition function τ, a reward function R, a mentor start state s_m, a number of samples N, and a constant $0 < \epsilon \leq 1$.

$R'(s) \leftarrow R(s) + p$ /* Modify reward structure */
$\pi'_m \leftarrow \text{PolicyIteration}\left(\tau, R'\right)$
$count \leftarrow 0$
for $i = 1$ to N **do**
  $h^i \leftarrow \text{SimPolicy}(\pi'_m)$ /* Simulate running π'_m */
  /* If sampled trajectory i passes through s */
  if $s \in h^i$ **then**
    $count \leftarrow count + 1$
  end if
end for
if $\frac{count}{N} \geq \epsilon$ **then**
  Accept /* Accept if more than ϵ of samples reach s */
else
  Reject
end if

Although Bayesian VOD determines the observer's pricing of mentor demonstrations, we note that policy regret can inform decisions about *which mentor* to ask for a demonstration, out of a collection of candidates. In this paper we show results for a single mentor and observer; however, for a set of mentors $\mathcal{M}$ and a single observer, we propose that the observer should select a mentor $m^* \in \mathcal{M}$ such that:

$$m^* = \underset{m \in \mathcal{M}}{\text{argmax}} \frac{1}{1-\delta} \left(\mathrm{E}_\theta \left[\hat{V}_m(s) \right] - \mathrm{E}_{\text{Dir}_o} \left[\hat{V}_o(s) \right] \right) \tag{14}$$

We intend to explore this topic as future work.

Results

In all the results we show, the observer recomputes its side payment offer at the beginning of each game, and recom-

putes its policy using policy iteration at the end of each game (after observing both its own experiences and those of the mentor, whether the mentor was helpful or not). For the observer to infer the actions taken by the mentor, the observer must explore. Our experiments use ϵ-greedy exploration with $\epsilon = 0.05$. Previous work (Price & Boutilier 2003) demonstrates that imitation remains a valuable learning technique even when more advanced methods of exploration are employed; we use ϵ-greedy exploration because it provides a straightforward baseline for future comparisons and its computational overhead is negligible.

Fig. 2(a) shows an example gridworld domain. In Fig. 2(b) we show cumulative reward over a series of 300 games (each one maximum 150 steps long) played in (a) under 3 experimental conditions. The black line shows cumulative reward using Algorithms 1 and 2 (with side payment amount deducted), and the gray line shows cumulative reward when the observer learns a path through ϵ-greedy exploration with an unhelpful mentor. Note that, without the inducement of side payments, the observer gravitates toward the low reward at lower right rather than the high reward at top achieved by the mentor.

In Fig. 2(c), we demonstrate δ-*sensitivity* when policy regret is used to determine the payoff amount (rather than Bayesian VOD). δ-sensitivity refers to the fact that, because policy regret decays toward 0 very slowly (for reasons discussed below), having a very high δ can lead to repeated, excessive side payments that negate the benefits of faster learning via imitation. Conversely, low δ means the observer can never tempt the mentor into offering a demonstration.

Fig. 2(d) shows examples of how Bayesian VOD and policy regret change with repeated demonstrations. Bayesian VOD drops off quickly; regret starts out near 0 (because our prior on $V_m(s)$ is 0), then asymptotes to a higher value as the mentor shows its competence. Note the difference in payoff scales shown vertically for each notion of VOD. Eventually, policy regret does decay to 0; this takes time, since the estimate is based on $\hat{V}_o(s)$, which is low until the observer learns to take actions that emulate the steps taken by the mentor. This point, at which following the mentor's policy becomes feasible, is shown as an inflection point in the cumulative reward graphs.

Fig. 3 shows a measure of *state relevance*, answering the question: from what states should a naive observer ask for demonstrations? Fig. 3(b) shows VOD after 20 demonstrations. The figure demonstrates a "frontier effect:" as the number of demonstrations around the mentor's path grows large, the value of demonstration on the path drops. Fig. 3(c) shows policy regret after 1 (left) and 20 (right) demonstrations from the state at lower left on the maze in Fig. 3(a); note the inverse relationship to (b). Fig. 3(d) shows the relevance for the observer of each state s. $Rel(s)$, at each point in the game, defined as:

$$Rel(s) = P(s|\pi_m(s_o))VOD(s) \quad (15)$$
$$= \mathrm{E}_{\tau,R}\left[VOD(s)\right] \quad (16)$$

where s_o is the observer's starting state. This definition selects states with high Bayesian VOD, weighted by the probability that the mentor's (optimal) policy π_m would carry

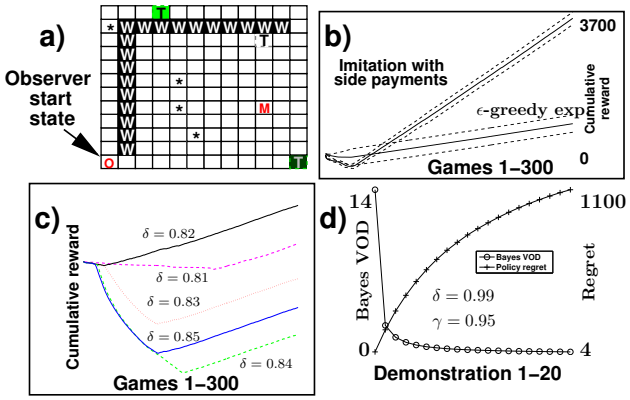

Figure 2: **Inducing demonstrations with side payments:** (a) 12×12 gridworld maze used in (b,d). "T" indicates terminal states; "W" are impassable walls; "$*$" are warps leading back to the start state at lower left. Lighter green squares have higher reward; white squares have small negative reward. Actions fail stochastically with probability 0.15. Observer starts at lower left (O); mentor starts midway at right (M). (b) Average over 5 random seeds of cumulative discounted reward of the observer over multiple games for the cases of a selfish mentor responding to side payments (black, upper line), adjusted to subtract side payments, and a control agent (gray line) using ϵ-greedy exploration with a mentor that does not respond to side payments. Dashed lines indicate $\pm$ 1 standard dev. (c) Example of δ-sensitivity using policy regret pricing on a simple 10×10 maze with no obstacles. Low δ leads to no demonstrations; high δ leads to excessive side payments because policy regret drops off slowly with time. (d) VOD of the start state (1,1) after repeated demonstrations of the optimal policy. Note difference in vertical scales for Bayes VOD vs. policy regret.

the observer to s if execution started from s_o. It suggests an answer to the question of when an observer should ask for help: if the observer reaches a state at which the relevance is higher than the VOI of exploring its own actions, it should ask for help. Note in this example that the relevance after 20 demonstrations concentrates on the frontier, near the state with small reward infrequently visited by the mentor (thus higher VOD). [2]

To complete our analogy with the Prisoner's Dilemma, we show the benefits of side payments in Fig. 4, a "machine shop" domain. Suppose there are 4 machines in a factory, numbered $1 \ldots 4$. Agent α knows the correct policy to operate machines 1 and 2; agent β knows how to operate machines 3 and 4. Operating a machine involves making state changes to some product being produced on the machine; if a mistake occurs in the process, the agent must start over again on the machine. Each agent starts each game on machine i with probability $P(i)$, and the agents know this distribution *a priori*. The agents have an incentive to cooperate: suppose agent α induces β to demonstrate operation of machine 3, then refuses to honor its side payment. Then β will refuse to reveal the correct policy for machine 4. A strong

[2]The probability $P(s|\pi_m(s_o))$ can be assessed by sampling from the Dirichlet distribution induced by the mentor history, or simply by the ML estimate, as shown here.

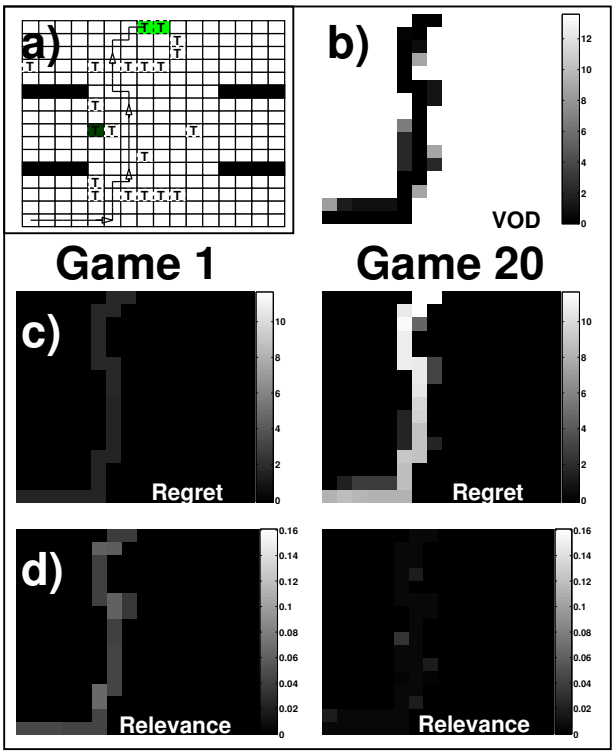

Figure 3: **Selecting relevant states for demonstration:** (a) 16×16 environment used to illustrate state relevance; demonstration starts at lower left. Mentor's policy is overlaid with arrows. (b) Bayesian VOD declines over time along the path demonstrated by the mentor. (c) State relevance $\mathrm{E}_{\hat{\tau},R}\left[VOD(s)\right]$ over time. Note how relevance becomes concentrated on a few states where demonstrations would be most valuable. (d) Policy regret rises along the path while the observer is still learning how to follow the mentor's policy.

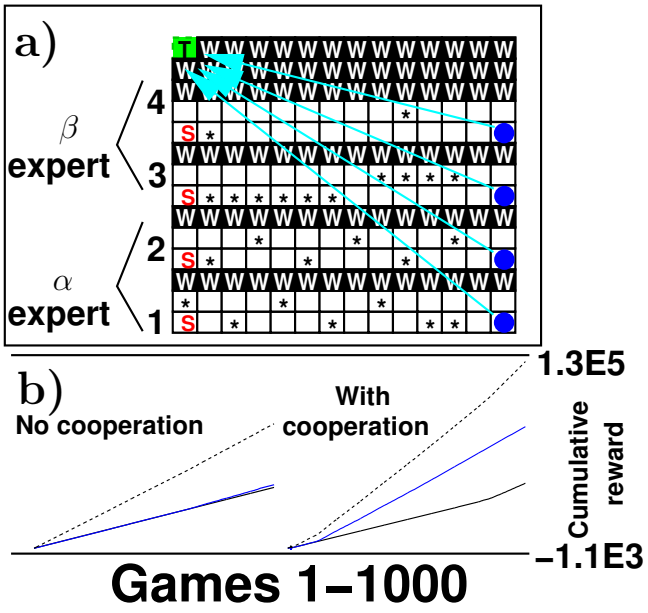

Figure 4: **Machine shop example:** (a) Problem domain. Machines are numbered $1 \ldots 4$. Each machine's start state is labeled "S"; after reaching the final state of the machine (blue circles at right), the operator receives a reward at top left. "$*$" squares reset the operator to the start state. State changes are modeled as a gridworld (so agents still choose from 4 actions on each time step); moving "East" along an assembly line represents making progress on the assembly line, until the final state (blue circle) is reached. (b) Payoff over time when the agents do (right) and do not (left) cooperate. Agent α's cumulative reward is shown as a black line; agent β's cumulative reward is shown as a blue line; side payments are taken into account. Combined reward of the two agents is dashed. After 1000 games, combined reward is approximately 56% greater with cooperation induced via side payments than without.

incentive thus exists for α to honor its agreement, particularly if the number of machines is small relative to the number of states in each machine. Fig. 4(a) shows the domain; machines are labeled $1 \ldots 4$ from top to bottom, and states within each machine proceed from left to right. "$*$" symbols represent mistakes that require the agent to start its operation on the machine from scratch. Fig. 4(b) shows the agents' cumulative rewards when cooperation is induced via side payments; lower cumulative rewards in (c) show the effects of learning when side payments are not used.

Conclusion

This paper introduces *active imitation learning*, or *learning by asking*, a potentially fruitful imitation learning paradigm covering cases where an observer can influence the frequency and the value of demonstrations that it is shown. In particular, this paper:

1. Applies active learning ideas to imitation learning, showing that value-of-information concepts can be adapted to imitative situations.
2. Proposes new algorithms for solving some important problems that arise in active imitation learning such as using side payments to induce imitation.
3. Provides simulation results demonstrating the applicability of active imitation learning to accelerating reinforcement learning.

Related problems are addressed in (Nunes & Oliveira 2003), where explicit information about which action to take can be transmitted, all agents employ the same action space, and all agents report their current cumulative rewards to one another, with perfect honesty, in a synchronized fashion. Explicitly transmitting knowledge about actions is unrealistic for many real-world cases, as in the learning tennis from observation example mentioned previously. As in (Price 2003), we disallow explicit communication about actions (all learning takes place through imitation via observation of mentor states only). Unlike (Price 2003), we assume a selfish mentor rather than a cooperative one. We do not claim to solve the general problem of setting side payments in general-sum multiagent imitation games to ensure cooperation; this is clearly a major open research problem. In the full multiagent case, observers have an incentive to cause multiple mentors to demonstrate trajectories, and mentors might need to have

expectations over other potential mentors' decisions. That is, mentors may find themselves in competition to garner side payments from the observer.

We have given a preliminary treatment to some problems in active imitation. Possible future topics include:

1. Extension of the framework to handle many agents simultaneously bartering for informational exchange;
2. Elaboration of the connections between game-theoretic equilibria and mentor-observer negotiations on the value of demonstration;
3. Domain-specific extensions to the VOD concept;
4. Implementation of active imitation on robotic platforms.

Acknowledgements

Supported by NSF grants IIS-0413335 and CCF-0622252, an ONR YIP award, and a Packard Fellowship to RPNR. We thank the anonymous reviewers for their helpful comments.

References

Abbeel, P., and Ng, A. Y. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proc. ICML*.

Axelrod, R. 2006. *The Evolution of Cooperation*. Perseus Books Group.

Dearden, R.; Friedman, N.; and Andre, D. 1999. Model based Bayesian exploration. In *Proc. UAI*, 150–159.

Dearden, R.; Friedman, N.; and Russell, S. J. 1998. Bayesian Q-learning. In *AAAI/IAAI*, 761–768.

DeGroot, M. H. 1986. *Probability and statistics*. Reading, MA: Addison-Wesley Publishing Co.

Dresher, M. 1961. *The Mathematics of Games of Strategy: Theory and Applications*. Prentice-Hall, Englewood Cliffs, NJ.

Howard, R. A. 1966. Information value theory. *IEEE Trans. System Science and Cybernetics* SSC-2:22–26.

Mannor, S.; Simester, D.; Sun, P.; and Tsitsiklis, J. N. 2004. Bias and variance in value function estimation. In *Proc. ICML*.

Nunes, L., and Oliveira, E. 2003. Cooperative learning using advice exchange. *Adaptive Agents and Multi-Agent Systems, LNCS* 2636:33–48.

Price, B., and Boutilier, C. 1999. Implicit imitation in multiagent reinforcement learning. In *Proc. ICML*, 325–334. Morgan Kaufmann, San Francisco, CA.

Price, B., and Boutilier, C. 2003. A Bayesian approach to imitation in reinforcement learning. In *Proc. IJCAI*, 712–717.

Price, B. 2003. *Accelerating Reinforcement Learning with Imitation*. Ph.D. Dissertation, University of British Columbia.

Schaal, S. 1997. Learning from demonstration. In *Advances in NIPS 9*.