Imitation Learning Using Graphical Models

Deepak Verma and Rajesh P.N. Rao

Dept. of Computer Science & Engineering
University of Washington
Seattle, WA, USA
{deepak,rao}@cs.washington.edu
http://neural.cs.washington.edu/

Abstract. Imitation-based learning is a general mechanism for rapid acquisition of new behaviors in autonomous agents and robots. In this paper, we propose a new approach to learning by imitation based on parameter learning in probabilistic graphical models. Graphical models are used not only to model an agent's own dynamics but also the dynamics of an observed teacher. Parameter tying between the agent-teacher models ensures consistency and facilitates learning. Given only observations of the teacher's states, we use the expectation-maximization (EM) algorithm to learn both dynamics and policies within graphical models. We present results demonstrating that EM-based imitation learning outperforms pure exploration-based learning on a benchmark problem (the FlagWorld domain). We additionally show that the graphical model representation can be leveraged to incorporate domain knowledge (e.g., state space factoring) to achieve significant speed-up in learning.

1 Introduction

Learning by imitation is a general mechanism for rapidly acquiring new skills or behaviors in humans and robots. Several approaches to imitation have previously been proposed (e.g., [1,2]). Many of these treat the problem of imitation as trajectory-following where the goal is to follow the teacher's trajectory as best as possible. However, imitation often involves the need to infer intentions and goals which introduces considerable uncertainty into the problem, besides the uncertainty already existing in the observation process and in the environment. Previous models of imitation have typically not been probabilistic and are therefore not geared towards handling uncertainty. There have been some recent efforts in modeling goal-based imitation [3] but these either assume that the dynamics of environment are given or need to learn the dynamics using a time-consuming exploration stage.

A different approach to imitation is based on ideas from the field of Reinforcement Learning (RL) [4]. In reinforcement learning, the agent is assumed to receive rewards in certain states and the agent's goal is to learn a state-to-action mapping ("policy") that maximizes the total future expected reward. The computational challenge of solving RL problem is hard for a variety of reasons: (1) the state space is often exponential in the number of attributes, and (2) for

J.N. Kok et al. (Eds.): ECML 2007, LNAI 4701, pp. 757–764, 2007.

uncertain environments with large state spaces, the agent needs to perform a large amount of exploration to learn a model of the environment before learning a good policy. These problems can be ameliorated by using imitation [5] (or "apprenticeship" [6]) where a teacher exhibits the optimal behavior that is observed by the student or the teacher guides the student to the most important states for exploration. Price and Boutilier formulate this in the RL framework as *Implicit Imitation* [7], in which the student learns the dynamics of the environment by passively observing the teacher without any explicit communication regarding what actions to take. This speeds up the learning of policies. However, these approaches rely on knowing or inferring an explicit reward function in the environment, which may not always be available or easy to infer.

In this paper, we propose a new approach to imitation that is based on probabilistic Graphical Models (GMs). We pose the problem of imitation learning as learning the *parameters* of the underlying GM for the mentor's and observer's behavior (we use the terms mentor/teacher (and observer/student) interchangeably in the paper). To facilitate the transfer of knowledge from mentor to observer we tie the parameters of dynamics for the mentor with that of the observer, and update the observer's policy using the learned mentor policy. Parameters are learned using the expectation-maximization (EM) algorithm for learning in GMs from partial data. Our approach provides a principled approach to imitation based *completely* on an internal GM representation, allowing us to leverage the growing number of efficient inference and learning techniques for GMs.

2 Graphical Models for Imitation

Notation: We use capital letters for variables and small case letters to denote specific instances. We assume there are two agents, the observer $\mathcal{A}^o$ and the mentor $\mathcal{A}^m$ operating in the environment[1]. Let Ω_S be the set of states in the environment and Ω_A the set of all possible actions available to the agent[2] (both finite). At time t, the agent is in state S_t and executes action A_t. The agent's state changes in a stochastic manner given by the transition probability $P(S_{t+1} \mid S_t, A_t)$, which is assumed to be independent of t, i.e., $P(S_{t+1} = s' \mid S_t = s, A_t = a) = \tau_{s's a}$. When obvious from context, we use s for $S_t = s$ and a for $A_t = a$, etc. For each state s and action a, there is a real valued reward $\mathcal{R}^m(s, a)$ for the mentor ($\mathcal{R}^o(s, a)$ for the observer) associated with being in state s and executing the action a (with negative values denoting undesirable states or the cost of the action). The parameters described above define a Markov Decision Process (MDP) [9]. Solving an MDP typically involves computing an optimal *policy* $a = \pi(s)$ that maximizes total expected future reward (either a finite

[1] We use the superscript to distinguish the two agents and omit it for common variables (e.g., dynamics of the environment).

[2] For simplicity of exposition, we assume that agents operate (non-interactively) in the *same* environment. However, as discussed in [8], this assumption is not essential and one can apply the techniques discussed here to the more general setting where observer and mentor(s) have different action and state spaces.

horizon cumulative reward or discounted infinite horizon cumulative reward) when action a is executed in state s.

In a typical Reinforcement Learning problem, the dynamics and the reward function are not known, and one cannot therefore compute an optimal policy directly. One can learn both these functions by exploration but this requires the agent to execute a large number of exploration steps before an optimal policy can be computed. Learning can be greatly sped up via *implicit imitation* [7] which involves an agent (the observer) observing another agent (mentor) who has similar goals. . The main idea is to allow the agent to quickly learn the parameters in the relevant portion of the state space, thereby cutting down on the exploration required to compute a near-optimal policy.

We assume that the mentor follows a stationary policy $\pi^m(s)$ which defines its behavior completely. The observer is only able to observe the sequence of states that mentor has been in ($S^m_{1:t}$) and *not the actions*: this is important because some of the most useful forms of imitation learning are those in which the teacher's actions are not available, e.g., when a robot must learn by watching a human – in such a scenario, the robot can observe body poses but has no access to the human's actions (muscle or motor commands). The task of the observer is then to compute the best estimate of the dynamics $\hat{\tau}$ and mentor policy $\hat{\pi}^m$, given its own history $S^o_{1:t}, A^o_{1:t}$ and the mentor's state history $S^m_{1:t}$. Note that π^m can be completely independent of the observer's reward function $\mathcal{R}^o$: in fact, the problem as formulated above does not require the introduction of a reward function at all. The goal is simply to imitate the mentor by estimating and executing the mentor's policy. In the special case where the mentor is optimizing the same reward function as the observer, π^m becomes the optimal MDP policy. Note that since the observer cannot see actions that the mentor took and the transition parameters are not given, the problem is different from other approaches which speed up RL via imitation [6,10].

2.1 Generative Graphical Model

Both the mentor and the observer are solving an MDP. One key observation we make is that *given* the mentor policy the action choice and dynamics can be modeled easily using a *generative model* based on the well-known graphical model for MDP shown in Fig. 1(a). One does not need to know the mentor's reward model as π^m completely explains the mentor state sequence observed. The figure shows the 2-slice representation of the *Dynamic Bayesian Network* (DBN) used to model the imitation problem. Since we are assuming that the two agents are operating in the same environment, they have the same transition parameters ($\tau^m{=}\tau^o{=}\tau$). Note that the two graphical models (for the mentor and observer respectively) are disconnected as the two agents are non-interacting. The mentor's actions are guided by the optimal mentor policy $P(A^m_t = a|S^m_t = s) = \pi^m(a|s)$ and the observer's actions by the policy $P(A^o_t = a|S^m_t = s) = \pi^o_t(a|s)$. Unlike the mentor, the observer updates its policy over time (hence the subscript t on π^o). We require only the mentor to have a stationary policy. The mentor observations $s^m_{1:T}$ are generated by "sampling" the DBN. In our

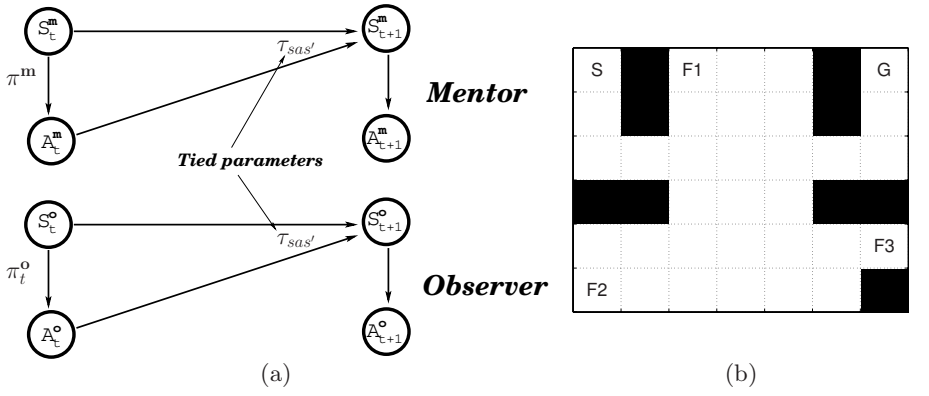


Fig. 1. Model and Domain for Imitation. (a) Graphical Model Representation for Imitation. (b) FlagWorld Domain.

experiments, when a goal state is reached, we jump to the start state in the next step. T thus represents the total number of steps taken by agent, which could span multiple "episodes" of reaching a goal state.

3 Imitation Via Parameter Learning

Our approach to imitation is based on estimating the unknown parameters $\theta = (\tau, \pi^m)$ of the graphical model in Fig. 1(a) given observed data as "evidence," i.e., $\hat{\theta} = (\hat{\tau}, \hat{\pi}^m) = \underset{\theta}{\text{argmax}}\, P(\theta | s^m_{1:T}, s^o_{1:T}, a^o_{1:T})$. Note that the evidence does *not* include mentor actions $A^m_{1:T}$. This means that the data is "incomplete" as not all nodes of the graphical model are observed. A well-known approach to learning the parameters of a GM from incomplete data [11] is to use the expectation-maximization (EM) algorithm [12]. Although any parameter learning method could be used, we use EM in the present study since it is a general-purpose, well-understood algorithm widely used in machine learning.

The EM algorithm involves starting with an initial estimate θ^0 (chosen randomly or incorporating any prior knowledge) which is then iteratively improved by performing the following two steps:

Expectation: The current set of parameters θ^i is used to compute a distribution (expectation) over the hidden nodes: $h(A^m_{1:T}) = P(A^m_{1:T} | \theta^i, s^m_{1:T}, s^o_{1:T}, a^o_{1:T})$. This allows the *expected sufficient statistics* to be computed for the complete data set.

Maximization: The distribution h is then used to compute the new parameters θ^{i+1} which maximize the (expected) log-likelihood of evidence:

$$\theta^{i+1} = \underset{\theta}{\text{argmax}} \sum_{a_{1:T}} h(a^m_{1:T}) \log(P(s^m_{1:T}, a^m_{1:T}, s^o_{1:T}, a^o_{1:T} | \theta))$$

When states and actions are discrete, the new estimate can be computed by simply using the expected counts. The two steps above are performed alternatively

until convergence. The method is guaranteed to improve performance in each iteration in that the incomplete log likelihood of data ($\log P(s^m_{1:T}, s^o_{1:T}, a^o_{1:T}|\theta^i)$) is guaranteed to increase in every iteration and converge to a local maximum [12]. We then use the estimate for $\hat{\theta}$ to control the observer. In particular, the observer combines the learned mentor policy $\hat{\pi}^m$ with an exploration strategy to arrive at the policy π^o_t.

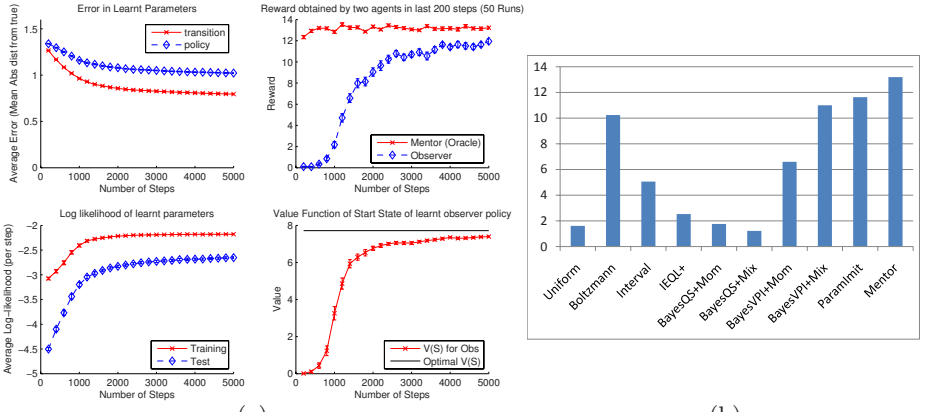
3.1 Parameter Learning Results

Domain: We tested our results on a benchmark problem known as the "Flag-World" domain [13] shown in Fig. 1(b). The agent's objective is to reach the goal state G starting from the state S and pick up a subset of the three flags located at states $F1$, $F2$ and $F3$. It receives a reward of 1 point for each flag picked up but rewards are discounted by a factor of $\gamma = 0.99$ at each time step until the goal is reached; the latter constraint favors shortest paths to goal. The environment is a standard maze environment used in RL [4] in that each action (N,E,S,W) takes the agent to the intended state with a high probability (0.9) and to a state perpendicular to the intended state with a small probability (0.1). The probability mass going into the wall or outside the maze is assigned to the state in which action taken. This domain is interesting in that there are 264 states (33 locations, augmented with a boolean attribute for each flag picked), resulting in a large number of parameters that needs to be learned (264×4 state action pairs for which $\tau(s, a, :)$ and $\pi^m(a|s)$ needs to be learned). However, the optimal policy path is sparse and hence only a small subset of parameters needs to be learned to compute a near-optimal policy, thereby making it ideal for demonstrating the utility of imitation as a medium to speed up RL.

Exploration versus Exploitation: We used the $\epsilon-$greedy method to trade-off exploration of the domain with exploitation of the current learned policy: a random action is chosen with probability ϵ, with ϵ gradually decreased over time to favor exploration initially and exploitation of the learned policy in later time steps.

Results: The results of EM-based learning are shown in Fig 2(a) (averaged over 50 runs). The parameters were learned in a "batch" mode where T was increased from 0 to 5000 in steps of 200 and reward in the last 200 steps was reported. Average reward received is shown in top right corner. Also shown are the Error in parameters (mean absolute difference w.r.t. true parameters[3]), the log-likelihood of the learned parameters and value function of start state under the current estimate for observer policy $V_{\hat{\pi}^o}(S)$ w.r.t the true transition parameters. The results show that the observer is able to learn the mentor policy to a high degree of accuracy, though not perfectly. The uncertain dynamics of the environment leads it to collect less rewards than the mentor as the optimal policy is not learned everywhere. An important point to note is that the error in

[3] The error between uniformly random parameters and true parameters is 1.5 for π^m and $\approx$1.75 for τ.



(a) (b)

Fig. 2. Imitation Learning Results for FlagWorld Domain. (a) (Clockwise) Error in parameters (mean absolute difference w.r.t. true parameters), average reward received, the log-likelihood of the learned parameters, and value function of start state $V_{\tilde{\pi}^o}(S)$ w.r.t the true transition parameters. (b) Comparison of learned policy (ParamImit) with some popular exploration techniques (measured in terms of average discounted reward obtained per 200 steps). ParamImit outperforms all the pure exploration-based methods.

parameters is still quite high even when observer policy is quite good, thereby confirming the intuition that only a small (relevant) subset of parameters needs to be learned well before the agent can start exploiting a learned policy.

Figure 2(b) compares the relative quality of the learned policy with a number of pure exploration-based techniques used in [13]. The bars represent the average discounted reward obtained per 200 steps in the 2nd stage, i.e., obtained in next 20,000 steps after an initial 1st stage of exploration consisting of 20,000 steps. For ParamImit (our algorithm) the average is taken after only 4000 steps of exploration. The rightmost bar is the Mentor value. As can be seen, ParamImit outperforms all the exploration strategies with far less experience.

3.2 Factored Graphical Model

A major advantage of using a graphical models-based approach to imitation is the ability to leverage domain knowledge to speed up learning. For example, the number of true parameters in the FlagWorld is actually much less than the number that was learned in the previous section since there are only 33 locations for which the transition parameters need to be learned: the dynamics are the same irrespective of which flags have been picked up. To reflect this fact, we can *factor* the mentor state S_t^m into location L_t^m and flag status variable "Picked Flag" PF_t^m as shown in Fig. 3(a) (and similarly for the observer). This reduces the number of transition parameters significantly (from $\tau_{sas'}$ to $\tau_{lal'}$).

We can incorporate domain knowledge about the flags by defining the CPT $P(PF_{t+1}|L_{t+1}, PF_t)$ as the ,

$$\begin{aligned} P(PF_{t+1}|L_{t+1}, PF_t) &= \delta(PF_{t+1}, pf(PF_t, i)) \quad \text{if } L_{t+1} = Fi \\ &= \delta(PF_{t+1}, PF_t) \quad \text{otherwise} \end{aligned}$$

where $pf(PF_t, i)$ is the *determinstic* function[4] which maps the old value of PF_t to one in which the i^{th} flag is picked up.

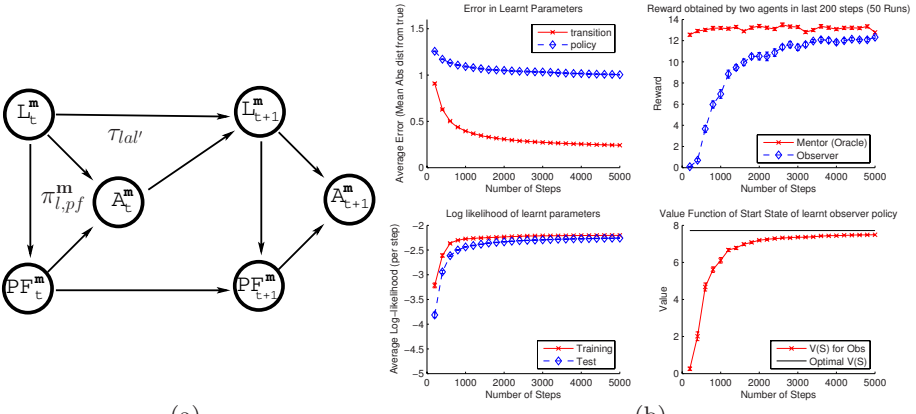


Fig. 3. Fast Learning using Factored Graphical Models. (a) Factored model for FlagWorld (only the mentor model is shown). (b) Results using factored model. Note the speed-up in learning w.r.t. the unfactored case (Fig. 2(a)).

The results of EM-based parameter learning for the factored graphical model are shown in Fig. 3(b). As expected, the error in transition parameters goes down much more rapidly than in the unfactored case (compare with Fig. 2(a)).

4 Conclusion

This paper introduces a new framework for learning by imitation based on modeling the imitation process in terms of probabilistic graphical models. Imitative policies are learned in a principled manner using the expectation-maximization (EM) algorithm. The model achieves transfer of knowledge by tying the parameters for the mentor's dynamics with those of the observer. Our results[5] demonstrate that the mentor's policy can be estimated directly from observations of

[4] This is a common trick used in GMs to encode *deterministic* domain knowledge.

[5] Additional results are presented in the extended version of the paper available at `http://neural.cs.washington.edu/`. In particular, we show how learning can be further sped up by incorporating reward information collected on the way. Also, we demonstrate the generality of parameter learning by extending the graphical model to learn task-oriented policies.

the mentor's state sequences and that significant speed-up in learning can be achieved by exploiting the graphical models framework to factor the state space in accordance with domain knowledge. Our current work is focused on testing the approach more exhaustively, especially in the context of robotic imitation. Not only do Graphical Models provide a computationally efficient framework for general imitation, they are also being used for modeling behavior [14]. An exciting prospect of using graphical models for imitation is the ease of extension to models with more abstraction, including partially observable, hierarchical, and relational models.

Acknowledgments

This material is based upon work supported by ONR, the Packard Foundation, and NSF Grants 0413335 and 0622252.

References

1. Schaal, S.: Is imitation learning the route to humanoid robots? Trends in Cognitive Sciences 3, 233–242 (1999)
2. Dautenhahn, K., Nehaniv, C.: Imitation in Animals and Artifacts. MIT Press, Cambridge, MA (2002)
3. Verma, D., Rao, R.P.N.: Goal-based imitation as probabilistic inference over graphical models. In: NIPS 18 (2006)
4. Sutton, R.S., Barto, A.: Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA (1998)
5. Atkeson, C.G., Schaal, S.: Robot learning from demonstration. In: Proc. 14th ICML, pp. 12–20 (1997)
6. Abbeel, P., Ng, A.Y.: Apprenticeship learning via inverse reinforcement learning. In: ICML '04, pp. 1–8 (2004)
7. Price, B., Boutilier, C.: Accelerating reinforcement learning through implicit imitation. JAIR 19, 569–629 (2003)
8. Price, B., Boutilier, C.: A bayesian approach to imitation in reinforcement learning. In: IJCAI, pp. 712–720 (2003)
9. Boutilier, C., Dean, T., Hanks, S.: Decision-theoretic planning: Structural assumptions and computational leverage. JAIR 11, 1–94 (1999)
10. Ratliff, N.D., Bagnell, J.A., Zinkevich, M.A.: Maximum margin planning. In: ICML06, pp. 729–736 (2006)
11. Heckerman, D.: A tutorial on learning with bayesian networks. Technical report, Microsoft Research, Redmond, Washington (1995)
12. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B 39, 1–38 (1977)
13. Dearden, R., Friedman, N., Andre, D.: Model-based Bayesian Exploration. In: UAI-99, San Francisco, CA, pp. 150–159 (1999)
14. Griffiths, T.L., Tenenbaum, J.B.: Structure and strength in causal induction. Cognitive Psychology 51(4), 334–384 (2005)