Multi-Agent Inverse Reinforcement Learning

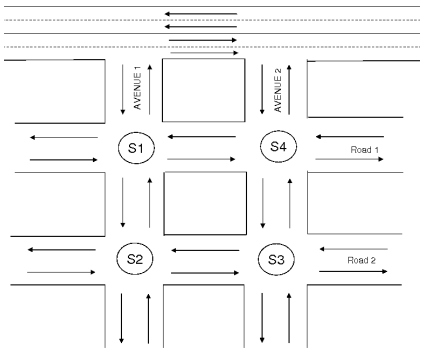
Sriraam Natarajan[1], Gautam Kunapuli[1], Kshitij Judah[2], Prasad Tadepalli[2], Kristian Kersting[3] and Jude Shavlik[1]
[1]Department of Biostat. & Med. Informatics, University of Wisconsin-Madison
[2]School of EECS, Oregon State University
[3]Fraunhofer IAIS, Germany

In traditional Reinforcement Learning (RL) [4], a single agent learns to act in an environment by optimizing some notion of long-term reward. However, there are several cases in which the reward function is not easily specifiable, or even known [3]. One such case where this occurs naturally is *apprenticeship learning* [1], in which an agent observes an expert's actions and has to learn from the demonstration. Consider, for example, a human operator who monitors traffic intersections and controls the signals to regulate the traffic. This expert's actions to determine optimal traffic regulation are guided by some reward function that is not explicitly observed. In order for an automated agent perform the same task, it has to observe the expert's demonstrations and learn a policy with respect to the true reward function, which is unknown. This is the goal of Inverse RL (IRL): to determine the reward function that the expert is optimizing. The observations include the expert's behavior over time in different situations, measurements of sensory inputs to the expert, and the model of the environment.

Thus far, IRL methods have usually been studied in the context of a single agent. Though this is an interesting problem that has deservedly received attention recently, there are several real-world scenarios in which *multiple* agents act independently to achieve a common goal. One such example is in Figure 1 (left): four agents control signals at four intersections based on local traffic density. The agents at intersections S_1 and S_4 are near a highway and their preferences will be different from those of the other two signals. While all the signals act in a locally optimal manner (i.e., they individually optimize the traffic at their own intersection), there is a necessity for co-ordination. It is conceivable, then, that there is a centralized controller that co-ordinates the actions of different agents so that they optimize the traffic over all the intersections.

We consider learning in situations similar to the scenario presented above, that is, multi-agent inverse reinforcement learning, a challenging problem for several reasons. A straightforward solution might be to consider individual agents and learn the reward functions for each agent individually; however, this does not always lead to a good global solution since the optimal behavior of one agent can possibly be suboptimal for another. Another challenge is that we may never be able to observe the actions of the centralized controller directly but only the actions of individual agents. The key idea here is to have the learning system observe individual agents, learn the reward functions of all them, and then control the agents' policies to optimize the *joint reward*. We consider weighting these agents, where some agents' policies are better optimized than others. For example, signals S_1 and S_4 are near a highway and hence the traffic entering (leaving) these two signals from (to) the highway needs to optimized. Our framework provides a natural way of incorporating such differences as weights for individual agents. We highlight two main contributions:

- We consider the multi-agent IRL problem in which agents co-ordinate to achieve a common goal. More precisely, we assume that it is possible to observe multiple agents for a significant period of time operating under a centralized mediator's policy. The controller's goal is to balance the policies of individual agents in order to maximize the (weighted) sum of the individual rewards. Thus, we try to determine the individual reward functions of the agents thus learning the reward function of the centralized controller. In order to achieve this goal, we consider the *average-reward setting* for IRL [2]. It has been shown that average-reward RL is more effective in many tasks where *discounted-reward* RL leads to myopic policies. Hence, a formal algorithm for



State	Expert Policy	Learner Policy
LHLH	a=4, p = 1	a=4, p= 0.3
MLLM	a=1, p = 1	a=1, p= 0.3
		a=4, p= 0.3
MMLM	a=1, p = 1	a=1, p= 0.3
		a =2, p=0.2
		a = 4, p=0.2
HMHL	a=1, p = 1	a=1, p= 0.43
		a=3, p=0.27
HHML	a=1, p = 1	a=1, p= 0.37
		a=3, p=0.27

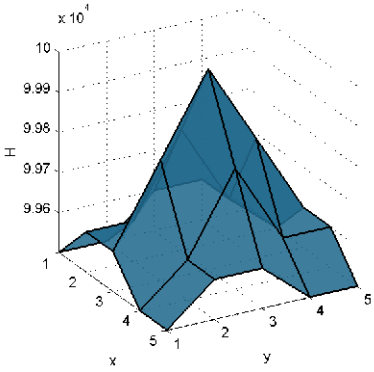


Figure 1: (**left**) A scenario in the traffic-routing domain. (**center**) Learned policy compared to the expert's policy for a few selected states on the traffic-routing domain. The expert always chooses actions deterministically, while the learner is able to choose from several actions that such that it gains a higher expected reward. This allows the learner to sometimes outperform the expert. Each agent has 4 actions and there are 4 agents, leading to 16 actions (a) and 4 components of the state for the central controller. (**right**) Learned value function for a 5×5 grid-world experiment.

inverse average-reward RL is crucial in solving several problems. Since we consider the multi-agent setting, the value functions, immediate rewards (correspondingly the average-rewards) all are vectors. We derive the equations for the IRL problem using the Bellman equations for average-reward RL which, when formulated as optimization problem lead to an LP or QP. We use different regularizations (L_1, L_2 and L_∞) to better understand their impact on the learned reward functions. Once the reward functions are learned, we use average reward multi-agent reinforcement learning algorithms to learn to act in the environment.

- The second contribution is the evaluation of the algorithms on various domains. First, in order to test the validity of our approach, we designed a simple 5×5 grid world problem where agents trying to reach the center of the grid are controlled by a hand-coded centralized controller. The learned value function is shown in Figure 1 (right). The value function forms a cone indicating that the cell at the center (goal) has the highest value and then decreases progressively as we move away from the center. The experiment serves as a proof-of-concept that our formulation can retrieve the true reward and value functions. The approach was also evaluated on the traffic-routing domain presented above: multiple traffic signals coordinated through a centralized controller. Here, given a few trajectories of the policies of different agents, we demonstrate that our approach learns reward functions that can imitate the expert policy reasonably and, in some cases, even improve upon the expert. This is desirable since it is not possible for the expert to provide extremely large number of trajectories that span the entire state space of the problem.

References

[1] P. Abbeel and A. Ng. Apprenticeship learning via inverse reinforcement learning. In *ICML*, 2004.

[2] S. Mahadevan and L. Kaelbling. Average reward reinforcement learning: Foundations, algorithms, and empirical results, 1996.

[3] A. Ng and S. Russell. Algorithms for inverse reinforcement learning. In *ICML*, 2000.

[4] R. Sutton and A. Barto. Reinforcement learning i: Introduction, 1998.

Topic: inverse reinforcement learning
Preference: oral/poster