Multi-agent Adversarial Inverse Reinforcement Learning with Latent Variables

Extended Abstract

Nate Gruver
Stanford University
ngruver@cs.stanford.edu

Jiaming Song
Stanford University
tsong@cs.stanford.edu

Mykel J. Kochenderfer
Stanford University
mykel@stanford.edu

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

ABSTRACT

We introduce an algorithm for inferring reward functions from expert human trajectories in multiagent environments. Current techniques exhibit poor sample-efficiency, lack stability in training, or scale poorly to large numbers of agents. We focus on settings with a large, variable number of agents and attempt to resolve these settings by exploiting similarities between agent behaviors. In particular, we learn a shared reward function using adversarial inverse reinforcement learning and a continuous latent variable. We demonstrate our algorithm on two real-world settings: traffic on highways and in terminal airspace.

ACM Reference Format:
Nate Gruver, Jiaming Song, Mykel J. Kochenderfer, and Stefano Ermon. 2020. Multi-agent Adversarial Inverse Reinforcement Learning with Latent Variables. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 3 pages.

1 INTRODUCTION

Accurate models of human behavior are increasingly important to safe and effective deployment of autonomous systems. Despite this need, behavior modeling remains difficult for various common problem settings. Urban environments, for example, still pose significant challenges for autonomous planning because of the uncertainty resulting from a high density of people [9]. To describe these scenarios robustly, a model must capture multi-modality in agent motivations and complex interactions that often scale superlinearly with the number of agents.

Two common approaches to modeling human behavior are imitation learning and inverse reinforcement learning (IRL). Imitation learning aims to produce trajectories that match a given expert distribution and can be attempted with techniques as simple as supervised learning – a.k.a. behavior cloning [6]. Inverse reinforcement learning, on the other hand, seeks to learn a reward function that can rationalize expert demonstrations [7]. The latter task is often much more challenging, but offers a concise description of the data generating process when accomplished. If the features used for IRL are sufficiently abstract, learned reward functions can also be transferred across problems [11].

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

1.1 Adversarial IRL (AIRL)

Inverse reinforcement learning (IRL [7, 8] seeks to identify the reward function under which the expert policies are "optimal". The MaxEnt RL framework relates the probability of sampling a trajectory by the optimal policy to the reward. With deterministic dynamics, T, the probability of sampling a trajectory $\tau = \{(s_t, a_t\}_{t=0}^{\infty}$ is determined by the following energy function:

$$p(\tau) = \frac{\exp(\sum_t r(s_t, a_t))}{Z(r)} \tag{1}$$

where $Z(r)$ is a normalization constant that sums over all valid trajectories. One can thus perform maximum likelihood estimation (MLE) of r over the demonstrations to recover the reward.

The normalization constant Z, however, is difficult to estimate for large or continuous state spaces. In adversarial inverse reinforcement learning [1, 2], adversarial training of a generator and discriminator is used to compute an importance sampling estimate of the normalization constant that can scale to high-dimensional problems.

1.2 Multi-agent IRL

In multi-agent environments, we can generalize the single reward function above to many reward functions $r^i \sim P_\phi(r)$. Our task thus becomes learning parameters ϕ such that sampled trajectories from all agents are indistinguishable from the set of expert trajectories.

To create a tractable objective, we assume the expert policies form a Logistic Stochastic Best Response Equilibrium (LSBRE, [10]). In LSBRE, imitation learning can be cast as maximum likelihood estimation over the expert trajectories as follows:

$$\max_\phi \mathbb{E}_{\tau\sim\pi_E}\left[\sum_{i=1}^{N} \log\left(\sum_{r^i} P_\phi(r^i)\frac{\exp(\sum_t r^i(s_t, a_t))}{Z(r^i)}\right)\right] \tag{2}$$

In the above problem, learning can be difficult as there are almost no structural assumptions among the agents, and thus the joint distribution over reward functions $P_\phi(r)$ can be arbitrarily complex. In particular, the objective in Equation 2 is difficult to optimize due to marginalization within the logarithm.

2 METHODS

2.1 Scalable Modeling with Latent Variables

To address the marginalization over $P_\phi(r)$ in Equation 2, we propose to model the policy and reward functions with latent variables.

Specifically, we may assume the reward function for agent i can be modeled using a latent variable z^i, and is defined as $r_\phi(a_t^i, s_t|z^i)$ with parameters ϕ respectively. We further assume that the latent variable has the prior $p(z) = \mathcal{N}(\mu_z, \sigma_z)$.

With the latent variables and conditional reward model, the objective in Equation 2 becomes:

$$\mathcal{L}(\phi) = \mathbb{E}_{\tau\sim\pi_E}\left[\sum_{i=1}^{N}\log\left(\int_{z^i} p(z^i)\frac{\exp(\sum_t r_\phi(a_t^i, s_t|z^i))}{Z(r_\phi, z^i)}\mathrm{d}z^i\right)\right] \quad (3)$$

To remove the summation over $p(z^i)$ in the log, we introduce a inference model $q_\omega(z^i|\tau^i)$, where $\tau^i = \{(s_t^i, a_t^i)\}$ is the trajectory for agent i. This leads to an evidence lower bound to $L(\phi)$ [3]:

$$\mathcal{L}(\phi) \geq \mathbb{E}_{\tau\sim\pi_E}\left[\sum_{i=1}^{N}\mathbb{E}_{z^i\sim q_\omega(z^i|\tau^i)}[\mathrm{ELBO}_{\phi,\omega}(\tau^i, z^i)]\right] \quad (4)$$

where $\mathrm{ELBO}_{\phi,\omega}(\tau^i, z^i)$ is defined as:

$$\sum_t r_\phi(a_t^i, s_t|z^i) - \log Z(r_\phi, z^i) - \log q_\omega(z^i|\tau^i) + \log p(z^i) \quad (5)$$

Given z^i, we can then optimize the first two terms in $\mathrm{ELBO}_{\phi,\omega}(\tau^i, z^i)$ with AIRL, which provides both the reward function and the corresponding policy as discussed next.

2.2 Multi-agent AIRL with Latent Variables

We propose an AIRL algorithm that maximizes the evidence lower bound objective in Equation 5. First, for the latent variable model q_ω, we introduce a inference network $q_\omega(z|\tau)$ that predicts the latent variable from trajectories. From Equation 5, this corresponds to the following objective:

$$\mathcal{L}_{q_\omega} = -\mathbb{E}_{\tau\sim\pi_E, z\sim q_\omega(z|\tau)}[\log q_\omega(z|\tau) - \log p(z)] \quad (6)$$

Then, conditioned on the latent variable z, we can transform the first term of Equation 5 using an AIRL approach. Here we need an additional discriminator $D_{\theta,\phi}(s, a, z)$ that depends on state s, action a and the latent variable z, whose goal is to discriminate generated trajectories and the demonstrations. Specifically, one provides a parameterized policy $\pi_\theta(a|s)$, and the discriminator is denoted as:

$$D_{\theta,\phi}(s, a, z) = \frac{\exp(r_\phi(s, a|z))}{\exp(r_\phi(s, a|z)) + \pi_\theta(a|s, z)} \quad (7)$$

The discriminator then minimizes the following objective:

$$\mathcal{L}_D = -\mathbb{E}_{\tau_E\sim\pi_E, z_E\sim q_\omega(z|\tau_E)}[\log D_{\theta,\phi}(s, a, z_E)] \quad (8)$$

$$-\mathbb{E}_{\tau\sim\pi_\theta(z), z\sim\mathcal{N}(0,1)}[\log(1 - D_{\theta,\phi}(s, a, z))] \quad (9)$$

$$-\mathbb{E}_{\hat{\tau_E}\sim\pi_\theta(z_E), z_E\sim q_\omega(z|\tau_E)}[\log(1 - D_{\theta,\phi}(s, a, z_E))] \quad (10)$$

where the first term encourages higher $D_{\theta,\phi}$ for demonstrations, and the second and third term encourages lower $D_{\theta,\phi}$ for trajectories generated by the policy when the latent variables are sampled from $p(z)$ or inferred from demonstrations.

The learned policy $\pi_\theta(a|s, z)$ produces an action distribution based on the latent variable and the current state, and its primary objective is to reach higher $D_{\theta,\phi}$ values. We use $\pi_\theta(z)$ for the shorthand notation for the policy $\pi_\theta(a|s, z)$ with latent variable z. To encourage the latent variables to be informative for generating the trajectories, we add a reconstruction loss such that trajectories in τ_E could be reconstructed via $q_\omega(z|\tau_E)$ (encoder) and $\pi_\theta(a|s, z)$ (decoder), similar to InfoGAIL [5]. This leads to the objective:

$$\mathcal{L}_G = -\mathbb{E}_{\tau\sim\pi_\theta(z), z\sim p(z)}[D_{\theta,\phi}(s, a, z)] \quad (11)$$

$$+\mathbb{E}_{\hat{\tau}_E\sim\pi_\theta(z_E), z_E\sim q_\omega(z|\tau_E), \tau_E\sim\pi_E}[\|\hat{\tau_E} - \tau_E\|_2] \quad (12)$$

These loss functions are iteratively minimized with stochastic gradient descent.

Table 1: Displacement of sampled trajectory from ground truth–both average over time and final displacement.

model	avg. disp. (m)	final disp. (m)
highD w/o latent	5.03	5.68
highD w/ latent	**4.93**	**5.61**
FAA w/o latent	670	808
FAA w/ latent	**596**	**785**

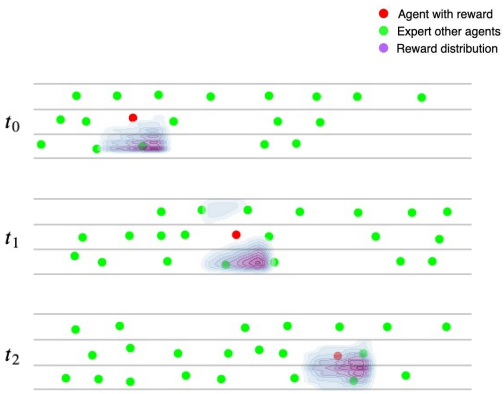

Figure 1: Visualization of learned reward function rolled out in time. Cars appear closer horizontally than in real life.

3 RESULTS

We tested our algorithm on highway trajectories from the highD dataset [4] and trajectories of aircraft in terminal airspace from non-public Federal Aviation Administration (FAA) data. Verifying learned reward function in an IRL setting is challenging when ground truth rewards are not available. We attempt to assay the reward functions by visualization and examining the corresponding policies.

Learned Policies

To demonstrate the plausibility of learned reward models, we generated trajectories from the corresponding policy distribution and measure displacements from the ground truth trajectories (Table 1). Video demonstrations of our learned policies are also available.[1]

Learned Reward Functions

We visualize learned rewards by plotting the distribution for nearby points. Figure 1 shows a visualization from the HighD dataset reflecting a lane-change objective.

[1] http://bit.ly/multi-agent-traffic

REFERENCES

[1] Chelsea Finn, Sergey Levine, and Pieter Abbeel. 2016. Guided Cost Learning: Deep Inverse Optimal Control via Policy Optimization. In *International Conference on Machine Learning (ICML)*. 49–58.

[2] Justin Fu, Katie Luo, and Sergey Levine. 2017. Learning Robust Rewards with Adversarial Inverse Reinforcement Learning. *arXiv preprint arXiv:1710.11248* (2017).

[3] Diederik P Kingma and Max Welling. 2013. Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114v10* (December 2013). arXiv:stat.ML/1312.6114v10

[4] Robert Krajewski, Julian Bock, Laurent Kloeker, and Lutz Eckstein. 2018. The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*.

[5] Yunzhu Li, Jiaming Song, and Stefano Ermon. 2017. Infogail: Interpretable imitation learning from visual demonstrations. In *Advances in Neural Information Processing Systems (NeurIPS)*. 3812–3822.

[6] Donald Michie, Michael Bain, and Jean Hayes-Michie. 1990. Cognitive models from subcognitive skills. In *Knowledge-based Systems for Industrial Contro*. Institution of Electrical Engineers, 71–99.

[7] Andrew Y. Ng and Stuart J. Russell. 2000. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning (ICML)*. 663–670.

[8] Stuart Russell. 1998. Learning agents for uncertain environments. In *Annual Conference on Computational Learning Theory (COLT)*. 101–103.

[9] Wilko Schwarting, Javier Alonso-Mora, and Daniela Rus. 2018. Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems* 1 (2018), 187–210.

[10] Lantao Yu, Jiaming Song, and Stefano Ermon. 2019. Multi-Agent Adversarial Inverse Reinforcement Learning. *arXiv preprint arXiv:1907.13220* (2019).

[11] Lantao Yu, Tianhe Yu, Chelsea Finn, and Stefano Ermon. 2019. Meta-Inverse Reinforcement Learning with Probabilistic Context Variables. *Advances in Neural Information Processing Systems (NeurIPS)* (2019).