Paper Study of Generative Adversarial Imitation Learning

Reinforcement Learning Theory Project by Meng-Huan Liu

June 2020

1 Introduction

Consider a specific setting of imitation learning where the problem is to learn to perform a task from expert demonstrations in which we are only given samples of trajectories from the expert. Before this paper was presented, there are two main approaches suitable for this problem setting: behavioral cloning and inverse reinforcement learning (IRL). For behavioral cloning which learns a policy as as supervised learning problem over state-action pairs from expert demonstrations, the main cons is that it tends to succeed at the cost of large amounts of expert demonstrations despite its simplicity. On the other hand, IRL, which finds a cost function under which the expert is uniquely optimal then finds an optimal policy with respect to this cost function via certain reinforcement learning procedure, are extremely expensive to run since its algorithm requires reinforcement learning in loops to learn the target cost function, making it difficult to scale to large environments.

In light of IRL, in which we learn a cost function that explains expert behavior but does not directly tell us what to act, while our true goal is often the latter one (to take actions imitating the expert). The authors want to find an algorithm that explicitly find how to act by directly learning a policy, bypassing any intermediate IRL step. The presented algorithm is also proved by the authors to be intimately connected to generative adversarial networks.

The method proposed in this paper is highly motivated by IRL but its performance is much better. From the results showcased in the paper, the model presented by the authors outperforms competing methods by a wide margin in complex and high-dimensional control tasks over various amounts of expert data.

2 Problem Formulation

In this section we first denote the notations and preliminary assumptions that we use in the following sections, followed by the main optimization problem of

our interests — a typical IRL problem, of which we will provide a new solutions to this problem different from the current one and also its proof in the next section.

Let $\overline{\mathbb{R}}$ denote the extended the union of real numbers and positive infinity. While the experiment of the algorithm runs in high-dimensional continuous environments, our proof works only with finite state and action space S and A. Π is the set of all stationary stochastic policies that take actions in A given states in S; successor states are drawn from the dynamics model P(s_0| s, a).We work in the γ-discounted infinite horizon setting, and we will use an expectation with respect to a policy $\pi \in \Pi$ to denote an expectation with respect to the trajectory it generates: [formulas to be inserted], where s_{00}, $a_t \sim \pi(\cdot|s_t)$, and $s_{t+1}(\cdot|s_t, a_t)$ for t $\geq$ 0. We will use $\hat{E}_\tau$ to denote an empirical expectation with respect to trajectory samples τ, and use π_E to refer to the expert policy.

Considering an IRL problem, where we are given an expert policy π_E that we wish to rationalize with. We assume the existence of solutions of maximum causal entropy IRL, which fits a cost function from a family of functions C with the optimization problem:

$$\underset{c \in C}{\text{maximize}} \left(\min_{\pi \in \Pi} -H(\pi) + \mathbb{E}_\pi[c(s,a)] \right) - \mathbb{E}_{\pi_E}[c(s,a)]$$

This optimization problem looks for a cost function $c \in C$ that assigns low cost to the expert policy and high cost to others, then we can find the expert policy by a certain reinforcement learning procedure:

$$\text{RL}(c) = \arg\min_{\pi \in \Pi} -H(\pi) + \mathbb{E}_\pi[c(s,a)]$$

which maps a cost function to high-entropy policies that minimize the expected cumulative cost.

From above, we know that our first step is to find cost function on the largest possible set of cost functions $C \in \{c : S \times A \to \mathbb{R}\}$, we want to tackle this problem by a way that both bypasses an intermediate IRL step (which is a loop) and is also suitable for large environments. For such a large set of C, to avoid overfitting given that we only have finite numbers of expert demonstrations, we incorporate a convex cost function regularizer $\psi : R^{S \times A} \to \overline{\mathbb{R}}$, then our regularized cost is:

$$\text{IRL}_\psi(\pi_E) = \arg\max_{c \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}} -\psi(c) + \left(\min_{\pi \in \Pi} -H(\pi) + \mathbb{E}_\pi[c(s,a)] \right) - \mathbb{E}_{\pi_E}[c(s,a)]$$

In the next section, we will solve this IRL problem in a new algorithm.

3 Theoretical Analysis

Let $\tilde{c} \in IRL_\psi(\pi_E)$. According to our second step in last section, we are interested in a policy given by running reinforcement learning on $\tilde{c}$, let's denote it RL($\tilde{c}$). To characterize this policy, we first define occupancy measure of a policy:

$$\rho_\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R} \text{ as } \rho_\pi(s,a) = \pi(a|s) \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi)$$

We can interpret the concept of occupancy measure as the time-discounted probability of state-action pairs that an agent encounters when taking policy π. Noted here that there is a constraint for occupancy measure: $\rho_{\pi_E} > 0$, we will see later that this guarantees the existence of $\tilde{c} \in IRL_\psi(\pi_E)$, in other word, guarantees that the IRL problem has a solution. We begin with **Proposition 3.1.**:

$$\pi_\rho(a|s) \triangleq \rho(s,a) / \sum_{a'} \rho(s,a')$$

where ρ is in the set of valid occupancy measure and π_ρ is the only policy whose occupancy measure is ρ. **This proposition states the one-to-one correspondence between a policy and its occupancy measure.**. Together with the concept of a convex conjugate for a function f:

$$f^*(x) = \sup_{y \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}} x^T y - f(y)$$

we can characterize $\mathrm{RL}(\tilde{c})$ as following **Proposition 3.2.**:

$$\mathrm{RL} \circ \mathrm{IRL}_{\psi}(\pi_E) = \arg\min_{\pi \in \Pi} -H(\pi) + \psi^*(\rho_\pi - \rho_{\pi_E})$$

We prove this proposition with the help of the following *Lemma 3.1*:

Lemma 3.1. *Let $\bar{H}(\rho) = -\sum_{s,a} \rho(s,a) \log(\rho(s,a)/\sum_{a'} \rho(s,a'))$. Then, $\bar{H}$ is strictly concave, and for all $\pi \in \Pi$ and $\rho \in \mathcal{D}$, we have $H(\pi) = \bar{H}(\rho_\pi)$ and $\bar{H}(\rho) = H(\pi_\rho)$.*

Let's denote $\overline{L}$ as follow:

$$\bar{L}(\rho, c) = -\bar{H}(\rho) - \psi(c) + \sum_{s,a} \rho(s,a) c(s,a) - \sum_{s,a} \rho_{\pi_E}(s,a) c(s,a).$$

Due to the convexity of $\overline{H}$ proved in *Lemma 3.1*, and also the convexity of ψ and $\overline{L}$ with respect to both c and ρ, we can wrap up the proof for **Proposition 3.2.** using minimax duality since the optimal cost function and policy form a saddle point of $\overline{L}$. IRL finds one coordinate of this saddle point, and running reinforcement learning on the output of IRL reveals the other coordinate.

By now, we can see that if we deal with the problem from the view of occupancy measure, then our goal becomes to seek a policy whose occupancy measure is close to the expert's measured by ψ^*. This also means that settings of ψ lead to various imitation learning algorithms that directly solve the optimization problem given by Proposition 321., including existing algorithm and our new one. To explain this, we start from the special case when ψ is a constant function, we are going to prove **Proposition 3.3** below:

If ψ is a constant function, $\tilde{c} \in \mathrm{IRL}_{\psi}(\pi_E)$, and $\tilde{\pi} \in \mathrm{RL}(\tilde{c})$, then $\rho_{\tilde{\pi}} = \rho_{\pi_E}$.

Similar to the proof of **Proposition 3.2**, let:

$$\bar{L}(\rho, c) = -\bar{H}(\rho) + \textstyle\sum_{s.a} c(s,a)(\rho(s,a) - \rho_E(s,a))$$

we have:

$$\begin{aligned}
\tilde{c} \in \mathrm{IRL}_{\psi}(\pi_E) &= \arg\max_{c \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}} \min_{\pi \in \Pi} -H(\pi) + \mathbb{E}_\pi[c(s,a)] - \mathbb{E}_{\pi_E}[c(s,a)] + \text{const.} \\
&= \arg\max_{c \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}} \min_{\rho \in \mathcal{D}} -\bar{H}(\rho) + \sum_{s,a} \rho(s,a) c(s,a) - \sum_{s,a} \rho_E(s,a) c(s,a) = \arg\max_{c \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}} \min_{\rho \in \mathcal{D}} \bar{L}(\rho, c)
\end{aligned}$$

the above problem is the dual problem of below problem with the Lagrangian $\overline{L}$, for which the costs c(s, a) serve as dual variables for equality constraints:

$$\underset{\rho \in \mathcal{D}}{\text{minimize}} -\bar{H}(\rho) \quad \text{subject to} \quad \rho(s,a) = \rho_E(s,a) \quad \forall\, s \in \mathcal{S}, a \in \mathcal{A}$$

So if there exists a solution $\tilde{c}$, then $\tilde{\rho} = \rho_E$

From the above proposition, if there were no cost regularization at all (same as constant regularization), the recovered policy will exactly match the expert's occupancy measure at all state-action pairs.However, it is intractable in large environments due to our finite set of expert demonstration samples and the big cost to use function approximation to learn a parameterized policy with a large number of constraints from dimension of $S \times A$. In succeeding paragraphs, we will show how occupancy measure matching becomes practical if we add some other regularizers thus leads to other algorithms. Let us first draw some conclusions from **Proposition 3.2** and **Proposition 3.3**. First, IRL is a dual of an occupancy measure matching problem, and the recovered cost function is the dual optimum. Previous IRL algorithms that solve reinforcement learning repeatedly in an inner loop can be viewed as repeatedly solves the primal problem (reinforcement learning) with fixed dual values (costs), which is ineffective. Second, the induced optimal policy is the primal optimum. The induced optimal policy is obtained by running RL after IRL, which is same as recovering the primal optimum from the dual optimum. Strong duality implies that this induced optimal policy is indeed the primal optimum, and therefore matches occupancy measures with the expert. So instead of viewing IRL as finding a cost function such that the expert policy is uniquely optimal, we convert the problem to induce a policy that matches the expert's occupancy measure from we have done above.

Next we work with regularizer ψ other than a constant, we modify last equation in the proof of **Proposition 3.2.** by adding smooth penalize term $d_\psi(\rho_\pi, \rho_E)$ denoting $\psi^*(\rho_\pi - \rho_E)$:

$$\underset{\pi}{\text{minimize}}\ d_\psi(\rho_\pi, \rho_E) - H(\pi)$$

Consider an apprenticeship learning algorithm, its optimizing objective is below:

$$\underset{\pi}{\text{minimize}}\ \max_{c \in \mathcal{C}} \mathbb{E}_\pi[c(s,a)] - \mathbb{E}_{\pi_E}[c(s,a)]$$

If we define indicator function δ_c, where δ_c(c)=0 if c $\in$ C and $+\infty$ otherwise, we can write the apprenticeship learning objective as:

$$\max_{c \in \mathcal{C}} \mathbb{E}_\pi[c(s,a)] - \mathbb{E}_{\pi_E}[c(s,a)] = \max_{c \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}} -\delta_{\mathcal{C}}(c) + \sum_{s,a}(\rho_\pi(s,a) - \rho_{\pi_E}(s,a))c(s,a) = \delta^*_{\mathcal{C}}(\rho_\pi - \rho_{\pi_E})$$

to recover the entropy term, we can simply scaling C by a constant α and takes it to infinity, then we can see that entropy-regularized apprenticeship learning is equivalent to performing RL following IRL with cost regularizer $\psi = \delta_c$.

After we found the underlying relationship between indicator regularizers for the linear cost function classes and existing apprenticeship learning algorithm,

which again prove that our theorem of dual problem form of IRL is true. We then propose the following new cost regularizer that combines the pros of the preceding two cases:

$$\psi_{\mathrm{GA}}(c) \triangleq \begin{cases} \mathbb{E}_{\pi_E}[g(c(s,a))] & \text{if } c < 0 \\ +\infty & \text{otherwise} \end{cases} \quad \text{where } g(x) = \begin{cases} -x - \log(1-e^x) & \text{if } x < 0 \\ +\infty & \text{otherwise} \end{cases}$$

The idea of this regularizer is that it places low penalty on cost functions c that assign an amount of negative cost to expert state-action pairs, otherwise heavy penalty. Compared with indicator regularizers mentioned before, ψ_{GA} is an average over expert data so that it can adapt to data while indicator regularizers cannot. The new regularizer satisfies the following property:

$$\psi^*_{\mathrm{GA}}(\rho_\pi - \rho_{\pi_E}) = \max_{D \in (0,1)^{\mathcal{S}\times\mathcal{A}}} \mathbb{E}_\pi[\log(D(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a))]$$

which is the optimal negative log loss of the binary classification problem of distinguishing between state-action pairs of π and π_E. After we shift it by a constant, we can obtain the Jensen-Shannon divergence:

$$D_{\mathrm{JS}}(\rho_\pi, \rho_{\pi_E}) \triangleq D_{\mathrm{KL}}\left(\rho_\pi \| (\rho_\pi + \rho_E)/2\right) + D_{\mathrm{KL}}\left(\rho_E \| (\rho_\pi + \rho_E)/2\right)$$

Then we can write a new imitation learning algorithm:

$$\underset{\pi}{\text{minimize}}\ \psi^*_{\mathrm{GA}}(\rho_\pi - \rho_{\pi_E}) - \lambda H(\pi) = D_{\mathrm{JS}}(\rho_\pi, \rho_{\pi_E}) - \lambda H(\pi)$$

which finds a policy whose occupancy measure minimizes Jensen-Shannon divergence to the expert's.
Finally, we present generative adversarial imitation learning to solve above equation for model-free imitation in large environments. Explicitly, we wish to find a saddle point (π, D) of the expression:

$$\mathbb{E}_\pi[\log(D(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a))] - \lambda H(\pi)$$

In light of generative adversarial networks, we first introduce function approximation for π and D: we will fit a parameterized policy and a discriminator network for them respectively. Then, we alternate between an Adam gradient step on D network, and a TRPO step on π network. The detail algorithm is as follow:

Algorithm 1 Generative adversarial imitation learning

1: **Input:** Expert trajectories $\tau_E \sim \pi_E$, initial policy and discriminator parameters θ_0, w_0
2: **for** $i = 0, 1, 2, \ldots$ **do**
3: Sample trajectories $\tau_i \sim \pi_{\theta_i}$
4: Update the discriminator parameters from w_i to w_{i+1} with the gradient

$$\hat{\mathbb{E}}_{\tau_i}[\nabla_w \log(D_w(s,a))] + \hat{\mathbb{E}}_{\tau_E}[\nabla_w \log(1 - D_w(s,a))] \tag{17}$$

5: Take a policy step from θ_i to θ_{i+1}, using the TRPO rule with cost function $\log(D_{w_{i+1}}(s,a))$. Specifically, take a KL-constrained natural gradient step with

$$\begin{aligned} &\hat{\mathbb{E}}_{\tau_i}\left[\nabla_\theta \log \pi_\theta(a|s) Q(s,a)\right] - \lambda \nabla_\theta H(\pi_\theta), \\ &\text{where } Q(\bar{s},\bar{a}) = \hat{\mathbb{E}}_{\tau_i}[\log(D_{w_{i+1}}(s,a)) \mid s_0 = \bar{s}, a_0 = \bar{a}] \end{aligned} \tag{18}$$

6: **end for**

4 Conclusion

The main limitation of this new algorithm is that it is not particularly sample efficient in terms of environment interaction during training. The number of such samples required to estimate the imitation objective gradient was comparable to the number needed for TRPO to train the expert policies from reinforcement signals. To improve learning speed, we can try initializing policy parameters with behavioral cloning, which requires no environment interaction at all.

Just like IRL, this approach does not interact with the expert during training, exploring randomly to determine which actions bring a policy's occupancy measure closer to the expert's. If we can interact with the expert, we can simply ask the expert for such actions. So if we are able to modify the method so that it can combines well-chosen environment models with expert interaction, then we can win in terms of sample complexity of both expert data and environment interactions.