A Note on Generative Adversarial Imitation Learning

Ruo-Lin Ko

Department of Computer Science

National Chiao Tung University

r0856123.cs08g@nctu.edu.tw

1 Introduction

Imitation learning is to learn a task from expert's demonstrations. There are two main approaches to do so, which are Behavioral cloning (BC) and Inverse reinforcement learning (IRL).

Behavioral cloning learns a policy using supervised learning. It optimizes a cost function between behavior and expert's state-action transition pairs. BC is simple but needs tons of expert's demonstrations, due to compounding error caused by covariate shift. In other words, if the agent encounters unknown states, it will easily fail to predict correct actions and leads to large compounding error.

Inverse reinforcement learning, on the other hand, learns a cost function, mostly a reward function. In this way, compounding error is no longer a problem. The cost function is learned by **maximum causal entropy IRL**. However, the IRL procedure requires a reinforcement learning in an inner loop to learn a cost function, this procedure is very expensive to run.

To tackle the expensive computation issue, instead of learning a cost function, the paper propose a method to directly learn policies from expert's demonstrations and bypassing any intermediate IRL step.

2 Problem Formulation

The paper starts from characterizing and adopting **maximum causal entropy IRL**. Given an expert policy π_E that we wish to rationalize with IRL, to fit an optimized cost function from a family of functions $\mathcal{C}$, it is an optimization problem:

$$\underset{c\in\mathcal{C}}{\text{maximize}} \left(\min_{\pi\in\Pi} -H(\pi) + \mathbb{E}_{\pi}[c(s,a)] \right) - \mathbb{E}_{\pi_E}[c(s,a)] \tag{1}$$

where $H(\pi) \triangleq \mathbb{E}_{\pi}[-\log \pi(a \mid s)]$ is the γ-discounted causal entropy of the policy π. The formula looks for a cost function that assigns low value to the expert policy while assigns high value to other policies. Such cost function can be found using a reinforcement learning procedure:

$$\mathrm{RL}(c) = \arg\min_{\pi\in\Pi} -H(\pi) + \mathbb{E}_{\pi}[c(s,a)] \tag{2}$$

Preprint. Under review.

3 Theoretical Analysis

First define c functions in $\mathcal{C}$ as $\mathbb{R}^{\mathcal{S}\times\mathcal{A}} = \{c : \mathcal{S}\times\mathcal{A} \to \mathbb{R}\}$. The transformations can be expressed using Gaussian process or neural networks without hand-crafted features. Since the large set of $\mathcal{C}$, IRL can easily overfit when provided a finite dataset. Therefore the authors add a convex cost function regularizer $\psi : \mathbb{R}^{\mathcal{S}\times\mathcal{A}} \to \bar{\mathbb{R}}$. Note that convexity is a not particularly restrictive requirement.

So now the IRL primitive procedure is defined as:

$$\mathrm{IRL}_\psi(\pi_E) = \arg\max_{c\in\mathbb{R}^{S\times\mathcal{A}}} -\psi(c) + \left(\min_{\pi\in\Pi} -H(\pi) + \mathbb{E}_\pi[c(s,a)]\right) - \mathbb{E}_{\pi_E}[c(s,a)] \tag{3}$$

Let $\tilde{c} \in \mathrm{IRL}_\psi(\pi_E)$. We are interested in a policy given by $\mathrm{RL}(\tilde{c})$ which is the policy learned by RL after learning the cost function from IRL.

Now the goal is to find the policy from $\mathrm{RL}(\mathrm{IRL}(\pi_E))$. The equation can be re-written as:

Proposition 3.1 $\mathrm{RL}\circ\mathrm{IRL}_\psi(\pi_E) = \arg\min_{\pi\in\Pi} -H(\pi) + \psi^*(\rho_\pi - \rho_{\pi_E})$

where occupancy measure ρ is the distribution of state-action pair under the policy and ψ^* is the convex conjugate of ψ. I will skip the proof of Proposition 1 here. If you are interesting, you can find it in Appendix A.1 at original paper. The Proposition 1 shows that the IRL procedure implicitly try to match the occupancy measures between a policy and the expert's policy, measured by the convex function ψ^*, so different ψ will result in different imitation learning.

There is a special case when the ψ is a constant function, $\rho_{\tilde{\pi}}$ will be equal to ρ_{π_E}. The following proof will try to derive that **IRL is a dual of an occupancy measure matching problem**. In other words, we can optimize the IRL procedure by leveraging an occupancy measure matching problem.

Corollary 3.1 *If ψ is a constant function, $\tilde{c} \in \mathrm{IRL}_\psi(\pi_E)$, and $\tilde{\pi} \in \mathrm{RL}(\tilde{c})$, then $\rho_{\tilde{\pi}} = \rho_{\pi_E}$*

To show this, we will need a lemma that lets us speak about causal entropies of occupancy measures:

Lemma 3.1 *Let $\bar{H}(\rho) = -\sum_{s,a}\rho(s,a)\log\left(\rho(s,a)/\sum_{a'}\rho(s,a')\right)$. 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 $\bar{H}(\rho) = H(\pi_\rho)$.*

The proof of this lemma is also in Appendix A.1. Proposition 3.1 and Lemma 3.1 together allow us to freely switch between policies and occupancy measures when considering functions involving causal entropy and expected costs, as in the following lemma:

Lemma 3.2 *If $L(\pi,c) = -H(\pi) + \mathbb{E}_\pi[c(s,a)]$ and $\bar{L}(\rho,c) = -\bar{H}(\rho) + \sum_{s,a}\rho(s,a)c(s,a)$, then for all cost functions c, $L(\pi,c) = \bar{L}(\rho_\pi,c)$ for all policies $\pi\in\Pi$, and $\bar{L}(\rho,c) = L(\pi_\rho,c)$ for all occupancy measures $\rho\in\mathcal{D}$*

Proof of Corollary 3.2.1. Define $\bar{L}(\rho,c) = -\bar{H}(\rho) + \sum_{s,a}c(s,a)\left(\rho(s,a) - \rho_E(s,a)\right)$. Given that ψ is a constant function, we have the following, due to Lemma 3.2:

$$\tilde{c} \in \mathrm{IRL}_\psi(\pi_E) = \arg\max_{c\in\mathbb{R}^{S}\times\mathcal{A}} \min_{\pi\in\Pi} -H(\pi) + \mathbb{E}_\pi[c(s,a)] - \mathbb{E}_{\pi_E}[c(s,a)] + \text{ const.} \tag{4}$$

$$= \arg\max_{c\in\mathbb{R}^{S}\times\mathcal{A}} \min_{\pi\in\Pi} -H(\rho) + \sum_{s,a}\rho(s,a)c(s,a) - \sum_{s,a}\rho_E(s,a)c(s,a) = \arg\max_{c\in\mathbb{R}^{S}\times\mathcal{A}} \min_{\pi\in\Pi} \bar{L}(\rho,c) \tag{5}$$

This is the dual of the optimization problem

$$\underset{\rho\in\mathcal{D}}{\text{minimize}} -\bar{H}(\rho) \quad \text{subject to} \quad \rho(s,a) = \rho_E(s,a) \quad \forall s\in\mathcal{S}, a\in\mathcal{A} \tag{6}$$

Now we have proven that IRL is a dual of an occupancy measure matching problem when ψ is a constant function. But it is not practically useful because the expert's demonstrations only as a finite set of samples. In large environment states, most of the expert's occupancy measure values will be exactly zero. In this case, the learned policy will never visit the unseen state-action pairs.

To tackle this problem, the authors relax Eq. (6) into the following form:

$$\text{minimize}_\pi \, d_\psi \left(\rho_\pi, \rho_E\right) - H(\pi) \tag{7}$$

where $d_\psi \left(\rho_\pi, \rho_E\right) \triangleq \psi^* \left(\rho_\pi - \rho_E\right)$, by modifying ψ to smoothly penalize the difference between the occupancy measures. With certain settings of ψ in Eq. (7), it can be transformed into the form of regularized variants of existing apprenticeship learning algorithms.

$$\underset{\pi}{\text{minimize}} \max_{c \in \mathcal{C}} \mathbb{E}_\pi[c(s,a)] - \mathbb{E}_{\pi_E}[c(s,a)] \tag{8}$$

The goal is to find the **policy** performing better than the expert policy. We know that Eq. (8) is a special case of Eq. (7) with a certain setting of ψ. Define the indicator function $\delta_\mathcal{C} : \mathbb{R}^{\mathcal{S} \times \mathcal{A}} \to \overline{\mathbb{R}}$, $\delta_\mathcal{C}(c) = 0$ if $c \in \mathcal{C}$ and $+\infty$ otherwise. We can write the apprenticeship learning Eq. (8) as:

$$\max_{c \in \mathcal{C}} \mathbb{E}_\pi[c(s,a)] - \mathbb{E}_{\pi_E}[c(s,a)] = \max_{c \in \mathbb{R}^{S \times \mathcal{A}}} -\delta_\mathcal{C}(c) + \sum_{s,a} \left(\rho_\pi(s,a) - \rho_{\pi_E}(s,a)\right) c(s,a) = \delta^*_\mathcal{C} \left(\rho_\pi - \rho_{\pi_E}\right) \tag{9}$$

Therefore, we see that entropy-regularized apprenticeship learning:

$$\text{minimize} -H(\pi) + \max_{c \in \mathcal{C}} \mathbb{E}_\pi[c(s,a)] - \mathbb{E}_{\pi_E}[c(s,a)] \tag{10}$$

This equation is equal to Proposition 3.1 when $\psi = \delta_\mathcal{C}$. The authors have shown that $\text{RL}(\text{IRL}(\pi_E))$ problem can be solved using apprenticeship learning. So what regularizer to choose now? The authors propose a regularizer that turns Eq. (10) into a GAN objective function:

$$\psi_{\text{GA}}(c) \triangleq \begin{cases} \mathbb{E}_{\pi_E}[g(c(s,a))] & \text{if } c < 0 \\ +\infty & \text{otherwise} \end{cases} \quad \text{where } g(x) = \begin{cases} -x - \log\left(1 - e^x\right) & \text{if } x < 0 \\ +\infty & \text{otherwise} \end{cases} \tag{11}$$

$$\psi^*_{\text{GA}} \left(\rho_\pi - \rho_{\pi_E}\right) = \max_{D \in (0,1)^{S \times A}} \mathbb{E}_\pi[\log(D(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a))] \tag{12}$$

where D is a discriminative classifier $D : \mathcal{S} \times \mathcal{A} \to (0,1)$. The choice of ψ^*_{GA} is derived from GAN and is proven in Appendix at original paper. The upper bound of Eq. (12) the optimal cost function distinguishes between state-action pairs of π and π_E. It turns out, that the optimal loss is similar to the Jensen-Shanon divergence:

$$\text{minimize}_\pi \, \psi^*_{\text{GA}} \left(\rho_\pi - \rho_{\pi_E}\right) - \lambda H(\pi) = D_{\text{JS}} \left(\rho_\pi, \rho_{\pi_E}\right) - \lambda H(\pi) \tag{13}$$

To find a saddle point (π, D) that solves Eq. (13), the authors propose an algorithm GAIL.

4 Conclusion

The authors propose an alternative perspective of IRL which bypassing the intermediate IRL steps. They prove that IRL is essentially a dual problem of occupancy measure problem. Furthermore, they

show that occupancy measure problem is equal to apprenticeship learning. In apprenticeship learning, they ultilize discriminator to replace the procedure of cost function. In this way, the overall IRL training turns to be a GAN training procedure.

In this paper, the objective function of GAIL is derived using Jensen-Shannon divergence D_{JS}. Everyone knows that JS divergence is tricky when the distributions are not overlapping. It can't distinguish the distances causing the network hard to converge. To tackle the issue, there are other metrics to compute distances between distributions such as Wasserstein metric, ...etc. It may have better performance if we try different metrics.