

Generative Adversarial Imitation Learning

Yeming Wen

2019-02-23

- ▶ **Definitions:** Action space \mathcal{A} and sample space \mathcal{S} . Π is the set of all policies. Also assume $P(s'|s, a)$ is the dynamics model. In this paper, π_E denotes the expert policy.
- ▶ **Imitation Learning:** Learning to perform a task from expert demonstrations without querying the expert while training.
- ▶ **Behavioral cloning:** Its success depends on large amounts of data.
- ▶ **Inverse RL:** The paper adopts the maximum causal entropy IRL which fits a cost function c with the following problem.

$$\begin{aligned}\pi^* &= \arg \min_{\pi \in \Pi} -H(\pi) + \mathbb{E}_{\pi}[c(s, a)] \\ \tilde{c} &= \arg \max_{c \in \mathcal{C}} \mathbb{E}_{\pi^*}[c(s, a)] - \mathbb{E}_{\pi_E}[c(s, a)]\end{aligned}$$

where $H(\pi) = \mathbb{E}_{\pi}[-\log \pi(a|s)]$ is the entropy of the policy.

- ▶ The reason why we want to maximize the entropy is we want to make the least claim of the model while fitting the data.
- ▶ IRL learns a cost function that prioritizes entire trajectories.
- ▶ It doesn't have compounding error which occurs when the only fits single-timestep decisions, such as behavioral cloning.
- ▶ However, IRL is generally expensive because it requires reinforcement learning in the inner loop.
- ▶ Learning a cost function doesn't tell the learner how to act (policy).
- ▶ We hope to build a new algorithm based on IRL which can lead to an induced policy.

- ▶ We first study the policies found by RL on costs learned by IRL on the largest possible set of cost functions $\mathcal{C} = \{c : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}\}$.
- ▶ Also need to define a convex cost function regularizer $\psi : \mathbb{R}_{\mathcal{S} \times \mathcal{A}} \rightarrow \mathbb{R}$, which turns out to be important in this paper.
- ▶ Re-write the Eq. 1 as the following:

$$\begin{aligned} IRL_{\psi}(\pi_E) = \arg \max_{c \in \mathcal{C}} & -\psi(c) + \left(\min_{\pi \in \Pi} -H(\pi) + \mathbb{E}_{\pi}[c(s, a)] \right) \\ & - \mathbb{E}_{\pi_E}[c(s, a)] \end{aligned}$$

- ▶ Define $RL(c) = \arg \min_{\pi \in \Pi} -H(\pi) + \mathbb{E}_{\pi}[c(s, a)]$.
Let $\tilde{c} \in IRL_{\psi}(\pi_E)$. We are interested in characterizing the induced policy $RL(\tilde{c})$.

- ▶ It is easier to characterize $RL(\tilde{c})$ if we transform optimization problems over policies into convex problems.
- ▶ So the paper introduces an occupancy measure $\rho_\pi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$:

$$\rho_\pi(s, a) = \pi(a|s) \sum_{t=0}^{\infty} \gamma^t P(s_t = s | \pi) \quad (1)$$

It can be interpreted as the distribution of state-action pairs when roll-out with policy π .

- ▶ There is an **one-to-one** correspondence between policy and occupancy measure. It also allows us to re-write the expected cost as

$$\mathbb{E}_\pi[c(s, a)] = \sum_{s, a} \rho_\pi(s, a) c(s, a) \quad (2)$$

- ▶ Lemma 1: If we define

$$\hat{H}(\rho) = - \sum_{s,a} \rho(s,a) \log(\rho(s,a) / \sum_{a'} \rho(s,a')) \quad (3)$$

then we have $\hat{H}(\rho) = H(\pi_\rho)$ and $H(\pi) = \hat{H}(\rho_\pi)$. So we can represent the entropy of a policy π with the occupancy measure ρ_π .

- ▶ Lemma 2: If we define,

$$\begin{aligned} L(\pi, c) &= -H(\pi) + \mathbb{E}_\pi[c(s, a)] \\ \hat{L}(\rho, c) &= -\hat{H}(\rho) + \sum_{s,a} \rho(s,a) c(s, a) \end{aligned}$$

then we have $L(\pi, c) = \hat{L}(\rho_\pi, c)$ and $\hat{L}(\rho, c) = L(\pi_\rho, c)$. The Lemma allows us to transform the problem from optimizing π to ρ .

Convex Conjugate

- ▶ Given a function f , it can be represented by the supremum of all affine functions that are majorized by f .
- ▶ For any given slope m , there may be many different constants b such that the affine function $\langle m, x \rangle - b$ is majorized by f . We only need the best such constant.
- ▶ That's what the convex conjugate f^* does. Given a slope m , f^* returns the best constant b such that $\langle m, x \rangle - b$ is majorized by f . Thus,

$$f^*(m) = \sup_x \langle m, x \rangle - f(x)$$

- ▶ Note that $f^{**} = f$.

- ▶ By Lemma 2, if ψ is a constant regularizer and $\tilde{c} \in IRL_{\psi}(\pi_E)$ and $\tilde{\pi} \in RL(\tilde{c})$, then $\rho_{\tilde{\pi}} = \rho_{\pi_E}$.
- ▶ Furthermore, we can also get the main result of the paper

$$RL \circ IRL_{\psi}(\pi_E) = \arg \min_{\pi \in \Pi} -H(\pi) + \psi^*(\rho_{\pi} - \rho_{\pi_E}) \quad (4)$$

where ψ^* is the convex conjugate of ψ , which is defined as

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

- ▶ It tells us that the ψ -regularized inverse RL seeks a policy whose occupancy measure is close to the expert's as measured by the convex function ψ^* .
- ▶ A good imitation learning algorithm boils down to a good choice of the regularizer ψ .

Occupancy Measure Matching

- ▶ As we showed previously, if ψ is a constant, then the resulting policy would have the same occupancy measures with expert at all states and actions.
- ▶ It is not practically useful because most of the occupancy measure of the expert values are exactly zero, due to the limited expert samples.
- ▶ Thus, exact occupancy measure matching will force the learned policy to never visit the unseen state-action pairs.
- ▶ If we restrict the class of cost function \mathcal{C} to be convex and set the regularizer ψ to be the indicator function of the set \mathcal{C} . Then optimization problem in (6) can be written as

$$\min_{\pi} -H(\pi) + \max_{c \in \mathcal{C}} \mathbb{E}_{\pi}[c(s, a)] - \mathbb{E}_{\pi_E}[c(s, a)] \quad (5)$$

which is a entropy-regularized apprenticeship learning problem.

Apprenticeship Learning

- Policy gradient method can be used to update the parameterized policy π_θ to optimize the apprenticeship objective, Eq. 7.

$$\begin{aligned}\nabla_\theta \max_{c \in \mathcal{C}} \mathbb{E}_{\pi_\theta} [c(s, a)] - \mathbb{E}_{\pi_E} [c(s, a)] &= \nabla_\theta \mathbb{E}_{\pi_\theta} [c^*(s, a)] \\ &= \mathbb{E}_{\pi_\theta} [\nabla_\theta \log \pi_\theta(a|s) Q_{c^*}(s, a)]\end{aligned}$$

where

$$c^* = \arg \max_{c \in \mathcal{C}} \mathbb{E}_{\pi_\theta} [c(s, a)] - \mathbb{E}_{\pi_E} [c(s, a)] \quad (6)$$

$$Q_{c^*}(\bar{s}, \bar{a}) = \mathbb{E}_{\pi_\theta} [c^*(\bar{s}, \bar{a}) | s_0 = \bar{s}, a_0 = \bar{a}] \quad (7)$$

- Fit c_i^* as defined above. Analytical solution is feasible if \mathcal{C} is restricted to Convex or Linear cost classes.
- Given the c_i^* , compute the policy gradient and take a TRPO step to produce $\pi_{\theta_{i+1}}$.

- ▶ Apprenticeship learning via TRPO is tractable in large environments but is incapable of exactly matching occupancy measures without careful tuning due to the restrictive cost classes \mathcal{C} .
- ▶ Constant regularizer ψ leads to exact matching but is intractable in large environments. Thus, GAIL is proposed to combine the best of both methods.

$$\psi_{GA}(c) \triangleq \begin{cases} \mathbb{E}_{\pi_E}[g(c(s, a))] & \text{if } c < 0 \\ +\infty & \text{otherwise} \end{cases}$$

where

$$g(x) = \begin{cases} -x - \log(1 - e^x) & \text{if } x < 0 \\ +\infty & \text{otherwise} \end{cases}$$

- ▶ The GAIL regularizer ψ_{GA} places low penalty on cost functions c that assign an amount of negative cost to expert state-action pairs; It heavily penalizes c if it assigns large cost to the expert.
- ▶ ψ_{GA} is an average over expert data so it can adjust to arbitrary expert datasets.
- ▶ In comparison, if ψ is an indicator function (Apprenticeship Learning), then it's always fixed.
- ▶ Another property of ψ_{GA} is its convex conjugate $\psi_{GA}^*(\rho_\pi - \rho_{\pi_E})$ can be derived in the following form:

$$\max_{D \in (0,1)^{S \times A}} \mathbb{E}_\pi[\log(D(s, a))] + \mathbb{E}_{\pi_E}[\log(1 - D(s, a))] \quad (8)$$

- ▶ It can be interpreted to find a discriminator that distinguishes trajectory between learned policy and expert policy. t

- Combining with the main result Eq. (6) in the paper,

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

The imitation learning problem is equivalent to find a saddle point (π, D) of the expression

$$\mathbb{E}_{\pi}[\log(D(s, a))] + \mathbb{E}_{\pi_E}[\log(1 - D(s, a))] - \lambda H(\pi) \quad (9)$$

- In terms of implementation, we just need to fit a parameterized policy π_{θ} with weights θ and a discriminator network $D_w : \mathcal{S} \times \mathcal{A} \rightarrow (0, 1)$ with weights w .
- Update D_w with Adam and update π_{θ} with TRPO iteratively.

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** $i = 0, 1, 2, \dots$ **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

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

- 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

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

- 6: **end for**
-

Results

