A Note on Policy Optimization With Demonstrations

Yi Cheng Shih
Department of Computer Science
National Yang Ming Chiao Tung University
z2x98653322@gmail.com

1 Introduction

Given a sparse reward environment, there's barely any guide for agent to explore the environment to find a way that leads to high rewards. Some of the papers overcome the problem by using expert demonstrations, but it needs the demonstrations to be of high-quality (already achieve high rewards).

This paper tries to let agent learn the expert's behavior at the early learning stage since reward sparsity. When the agent learn some skill, say, it already knows how to get high rewards to some extent, then it can rely on the policy generated experience to explore and learn. Note that the expert here does not necessarily need to provide high-quality demonstrations.

2 Problem Formulation

» Preliminaries and notations

The way to evaluate the performance of a policy π

$$\eta(\pi) = E_\pi[r(s,a)] = E_{(s_0,a_0,s_1,\ldots)}[\sum_{t=0}^{\infty} \gamma^t r(s,a)] \tag{1}$$

Occupancy measure says that given a state-action pair, if it's more likely to happen under the policy π, then the measured value will be higher

$$\rho_\pi(s) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi) \tag{2}$$

$$\rho_\pi(s,a) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi)\pi(a|s) \tag{3}$$

Given the occupancy measure, the objective η can be rewritten as

$$\eta(\pi) = \sum_{t=0}^{\infty} \sum_{s} P(s_t = s|\pi) \sum_{a} \pi(a|s)\gamma^t r(s,a) \tag{4}$$

$$= \sum_{s,a} \rho_\pi(s,a) r(s,a) \tag{5}$$

Surrogate function of $\eta(\pi)$

$$J_{\pi_{old}}(\pi) = \eta(\pi_{old}) + \sum_{s} \rho_{\pi_{old}}(s) \sum_{a} \pi(a|s) A_{\pi_{old}}(s,a) \tag{6}$$

from "Trust Region Policy Optimization paper" theorem 1. we have the following theorem

Theorem 2.1

$$\eta(\pi) \geq J_{\pi_{old}}(\pi) - CD_{KL}^{max}(\pi_{old}, \pi) \tag{7}$$

where

$$C = \frac{4\gamma\epsilon}{(1-\gamma)^2} \tag{8}$$

where

$$\epsilon = max_{s,a}|A_{\pi}(s,a)| \tag{9}$$

let $M_i(\pi) = J_{\pi_i}(\pi) - CD_{KL}^{max}(\pi_i, \pi)$, then we have

$$\eta(\pi_{i+1}) \geq M_i(\pi_{i+1}) \tag{10}$$

$$\eta(\pi_i) = J_{\pi_i}(\pi_i) = M_i(\pi_i) \tag{11}$$

$$\begin{aligned} \eta(\pi_{i+1}) - \eta(\pi_i) &\geq M_i(\pi_{i+1}) - M_i(\pi_i) && (12)\\ &= J_{\pi_i}(\pi_{i+1}) - J_{\pi_i}(\pi_i) - CD_{KL}^{max}(\pi_i, \pi_{i+1}) && (13) \end{aligned}$$

» Assumption

Initially, action a_E sampled from expert policy π_E is better than action a sampled from agent policy π for any state s.

$$E_{a_E \sim \pi_E, a \sim \pi}[A_{\pi}(s, a_E) - A_{\pi}(s, a)] \geq \delta \tag{14}$$

» Objective

A set of trajectories generated by expert policy π_E is denoted as

$$D^E = \{\tau_1, \tau_2, ..., \tau_N\} \tag{15}$$

We want to leverage D^E to maximize the objective η in (1) or equivalently (5)

objective function:

$$\mathcal{L}(\pi_\theta) = -\eta(\pi_\theta) + \lambda_1 D_{JS}(\pi_\theta, \pi_E) \tag{16}$$

3 Theoretical Analysis

Theorem 3.1 *Let*

$$\alpha = D_{KL}^{max}(\pi_{old}, \pi) = max_s D_{KL}(\pi(\cdot|s), \pi_{old}(\cdot|s))$$

$$\beta = D_{JS}^{max}(\pi_E, \pi) = max_s D_{JS}(\pi(\cdot|s), \pi_E(\cdot|s))$$

$$\epsilon_\pi = max_{s,a}|A_{\pi}(s,a)|$$

$$\epsilon_E = max_{s,a}|A_{\pi_E}(s,a)|$$

then

$$\eta(\pi) \geq J_{\pi_{old}}(\pi) - \frac{2\gamma(4\beta\epsilon_E + \alpha\epsilon_\pi)}{(1-\gamma)^2} + \frac{\delta}{1-\gamma} \tag{17}$$

let $M_i(\pi) = J_{\pi_i}(\pi) - C_{\pi_E} D_{JS}^{max}(\pi_E, \pi) - C_\pi D_{KL}^{max}(\pi_i, \pi) + \hat{\delta}$ = RHS of (17)

then

$$\eta(\pi_{i+1}) \geq M_i(\pi_{i+1}) \tag{18}$$

$$\eta(\pi_i) = J_{\pi_i}(\pi_i) = M_i(\pi_i) + C_{\pi_E} D_{JS}^{max}(\pi_E, \pi) - \hat{\delta} \tag{19}$$

$$\eta(\pi_{i+1}) - \eta(\pi_i) \geq M_i(\pi_{i+1}) - M_i(\pi_i) - C_{\pi_E} D_{JS}^{max}(\pi_E, \pi) + \hat{\delta} \tag{20}$$

$$= [J_{\pi_i}(\pi_{i+1}) - J_{\pi_i}(\pi_i) - C_\pi D_{KL}^{max}(\pi_i, \pi_{i+1})] - C_{\pi_E} D_{JS}^{max}(\pi_E, \pi_{i+1}) + \hat{\delta} \tag{21}$$

compare with (12), the additional part is $-C_{\pi_E} D_{JS}^{max}(\pi_E, \pi_{i+1}) + \hat{\delta}$, means that if JS divergence between π_E and π_{i+1} could be small enough, then the performance difference can be ($\sim \hat{\delta}$) higher than vanilla policy gradient methods.

We want to minimize the objective function (16) but we don't know π_E

Lemma 3.2 *$\pi_\rho(a|s) \triangleq \frac{\rho(s,a)}{\sum_{a'} \rho(s,a')}$, then π_ρ is the only policy whose occupancy measure is ρ*

By Lemma 3.2, objective function (16) can be written as

$$\begin{aligned} \mathcal{L}(\pi_\theta) &= -\eta(\pi_\theta) + \lambda_1 D_{JS}(\pi_\theta, \pi_E) \\ &= -\eta(\pi_\theta) + \lambda_1 D_{JS}(\rho_\theta, \rho_E) \end{aligned} \tag{22}$$

It's hard to minimize JS divergence directly

Theorem 3.3 *let $h(u) = log(sigmoid(u))$, $\bar{h}(u) = log(1 - sigmoid(u))$ and $U(s,a) : S \times A \to R$ be an arbitrary function, then*

$$D_{JS}(\rho_\pi, \rho_E) \geq sup_U(E_{\rho_\pi}[h(U(s,a))] + E_{\rho_E}[\bar{h}(U(s,a))]) + log4 \tag{23}$$

let h(U(s, a)) be an binary classifier(or in the terminology of GAN, a discriminator D) followed by a log, then we have

$$D_{JS}(\rho_\pi, \rho_E) \geq sup_D(E_{\rho_\pi}[log(D(s,a))] + E_{\rho_E}[log(1 - D(s,a))]) + log4 \tag{24}$$

D_{JS} is getting lower
iff
ρ_π is more similar to ρ_E
iff
RHS of (24) getting lower, when D is fixed
(assume that D(s, a) means the probability of (s, a) pair predicted to be from ρ_π, although it's quite counterintuitive, but otherwise it doesn't make sense in the paper.)

so instead of directly minimize JS divergence, we substitute RHS of (24) into objective (16)

$$\mathcal{L}(\pi_\theta) = -\eta(\pi_\theta) + \lambda_1 sup_D(E_{\rho_\pi}[log(D(s,a))] + E_{\rho_E}[log(1 - D(s,a))]) \tag{25}$$

To avoid overfitting, we add causal entropy $-H(\pi_\theta) = -E_\pi[-log\pi_\theta(a|s)]$ into the objective, which encourage the agent to be less deterministic.

$$\mathcal{L}(\pi_\theta) = -\eta(\pi_\theta) - \lambda_2 H(\pi_\theta) + \lambda_1 sup_D(E_{\rho_\pi}[log(D(s,a))] + E_{\rho_E}[log(1 - D(s,a))]) \tag{26}$$

then the optimization problem becomes like

$$min_\theta max_w - \eta(\pi_\theta) - \lambda_2 H(\pi_\theta) + \lambda_1(E_{\pi_\theta}[log(D_w(s,a))] + E_{\pi_E}[log(1 - D_w(s,a))]) \quad (27)$$

let $r'(s,a) = r(s,a) - \lambda_1 log(D_w(s,a))$

then (27) can be written as

$$min_\theta max_w - E_{\pi_\theta}[r(s,a) - \lambda_1 log(D_w(s,a))] - \lambda_2 H(\pi_\theta) + \lambda_1 E_{\pi_E}[log(1 - D_w(s,a))] \quad (28)$$

$$\equiv min_\theta max_w - E_{\pi_\theta}[r'(s,a)] - \lambda_2 H(\pi_\theta) + \lambda_1 E_{\pi_E}[log(1 - D_w(s,a))] \quad (29)$$

r' is the reshaped reward, in addition to r, it also consider expert's advice, especially when r(s, a) = 0 (reward sparsity), θ learns to behave like expert, so that there's a efficient exploration direction for agent to quickly gain some insight, and based on that to further maximize the expected reward.

Note:
typo: second last line of second last paragraph of Introduction section: "int terms of" -> "in terms of"

4 Conclusion

While an agent is learning how to behave like the expert, it' also exploring near the expert policy, it's helpful in the early stage of a sparse environment. But after the early stage(say, π outperforms π_E), it seems like keep considering the JS divergence between π_E and π doesn't make sense and may limit the exploration range and also waste computation on reward shaping and discriminator training.

If we use Behavior Cloning to learn π_E and train π_E to be better when time goes by, then use it to generate a new set of trajectories D^E, will it helps?