Policy Optimization with Demonstrations

Po Wei, Pan
Department of Power Mechanical Engineering
National Tsing Hua University
peter3354152@gmail.com

1 Introduction

- The main research challenges tackled by the paper :
  This research is produced to solve the problem of sparse reward task. First, they introduced several "Learning from demonstrations" method, such like replay memory based DQFD, DDPGFD, and some method based on reward reshaping or using demonstration data to shape the value function. Another method that use the concept of GAN is GAIL. Although these methods has reach some successful performance, but they also has some serious problems and drawbacks:

  - Because the previous method treat the demonstration data as self-generated data, so it might suffer the problem of under-exploiting demonstration data.
  - Some problems of LFD would rise when only imperfect demonstration data are given

  So this paper provide a state of the art method that can conquer these problem by create a simple dynamic reward reshaping based optimization algorithm – POFD.

- The high-level technical insights into the problem of interest :

- The main contributions of the paper (compared to the prior works) :
  POfD is the first one that can acquire knowledge from few and imperfect demonstration data to aid exploration in environments with sparse feedback.
- Your personal perspective on the proposed method :
  I think that this paper provide a huge improvement of Learn from demonstration method since it solve lots of problems. The most interest thing of this method is that it using the JS Divergence to calculate the difference of self-generated data and demonstration data. Also it use the concept of GAN to create a reward reshaping mechanism which is really interesting.

2 Problem Formulation

- Your notations (e.g. MDPs, value functions, function approximators,...etc) :
  For this papaer, it also came up with the MDP which is defined by a tuple (S, A, P, r, γ), and the expected discounted reward $\eta(\pi) = \mathbb{E}_\pi[r(s, a)]$ is used to evaluate the performance of the policy π. Following the standard definitions, the value function V_π and action value function Q_π can be written as $V_\pi(s) = \mathbb{E}[r(.,.)|s_0 = s]$ and $Q_\pi(s, a) = \mathbb{E}[r(.,.)|s_0 = s, a_0 = a]$. The advantage function $A_\pi(s, a) = Q_\pi(s, a) - V_\pi(s)$ reflects the expected additional reward that the agent will get after taking action a in state s.
- The optimization problem of interest :
  Reinforcement Learning (RL) is a set of algorithms trying to infer a policy achieving maximal reward $\eta(\pi)$ with the trajectory $\tau = (s_0, a_0), (s_1, a_1), ..., (s_T, a_T)$.

- The technical assumptions:

  For this paper it has some important assumptions and lemmas. First one is the *Occupancy measure*.It characterizes the distribution of action-state pairs when executing policy π:

  Definition 1. (*Occupancy measure*)

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

  $\rho_\pi(s, a) = \rho_\pi(s)\pi(a|s)$ is the occupancy measure of policy π

  Substituting $\rho_\pi(s, a)$ into the equation of expected discounted reward $\eta(\pi) = \mathbb{E}_\pi[r(s, a)]$, we can get:

  $$\begin{aligned}\mathbb{E}_\pi[r(s,a)] &= \sum_{t=0}^{\infty} \sum_s P(s_t = s|\pi) \sum_a \pi(a|s)\gamma^t r(s,a)) \\ &= \sum_s \rho_\pi(s) \sum_a \pi(a|s) r(s,a) \\ &= \sum_{s,a} \rho_\pi(s,a) r(s,a)\end{aligned}$$

  Lemma 1.

  Suppose ρ is the occupancy measure for $\pi_\rho(a|s) \equiv \rho(s,a)/\sum_{a'} \rho(s,a')$ Then π_ρ **is the only policy whose occupancy measure is** ρ.

  With the lemma, the occupancy measure could uniquely specifies a policy which is important for the analysis.

- Preliminaries :

  In practice, most RL tasks and environments do not provide a well-designed reward function. Instead, only a little sparse feedback indicating whether the goal is reached is available.In this work, they solve the LFD problem by developing a method capable of learning from both (sparse) rewards and expert demonstrations. And there's also an important assumption property of demonstration data:

  Assumption 1.

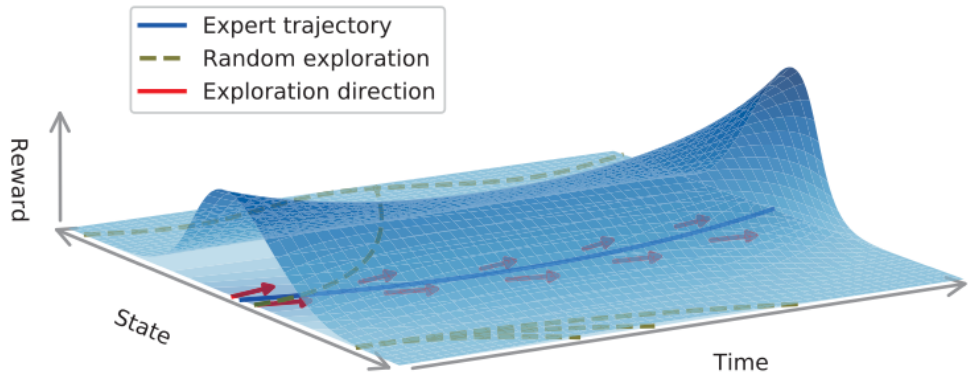
  In early learning stages, they assume expert policy π_E will provide higher advantage value with a margin as least δ over current policy π:

  $$\mathbb{E}_{aE\pi E, a\,\pi}[A_\pi(s, a_E) - A_\pi(s, a)] \geq \delta$$

  It didn't assume the expert policy π_E to be advantageous over all the policies, the proposed POfD will learn from both rewards and demonstrations and can possibly learn a policy better than π_E through exploration on its own in later learning stages.

3 Theoretical Analysis

For the paper invented method "Policy Optimization with Demonstrations".We usually start optimizing the policy π_θ by random exploration, which may cause slow convergence when the state and action spaces have high dimensionality, and may even lead to failure when the environment feedback is sparse. The main purpose is to forcing the policy to explore in the nearby region of the expert policy π, which is specified by several demonstrated trajectories D_E. Just like the following figure:



They use the Jensen-Shannon Divergence to be the demonstration-guided exploration term:

$$L_M(\pi_\theta) = D_{JS}(\pi_\theta, \pi_E)$$

And this give a new learning objective:

$$L_M(\pi_\theta) = -\eta(\pi_\theta) + \lambda_1 D_{JS}(\pi_\theta, \pi_E)$$

λ_1 is a trading-off parameter. The important point is the above policy divergence measure between π_θ and π_E is infeasible as π_E is unknown. **So here we use the lemma 1. We can instead define a divergence over the occupancy measures $\rho_\pi(s,a)$ and $\rho_{\pi E}(s,a)$, which is easier to optimize.** So we can rewrite the learning objective to

$$L_M(\pi_\theta) = -\eta(\pi_\theta) + \lambda_1 D_{JS}(\rho_\theta, \rho_E)$$

- **Benefits of Exploration with Demonstrations**:
  This paper also proof that the guiding term could boosts the advantage value for the learned policy, and brings non-trivial benefits in terms of policy improvement for policy gradient methods.

Algorithm 1 Policy optimization with demonstrations

Input: Expert demonstrations $\mathcal{D}_E = \{\tau_1^E, \ldots, \tau_N^E\}$, initial policy and discriminator parameters θ_0 and w_0, regularization weights λ_1, λ_2, maximal iterations I.
for $i = 1$ to I **do**
  Sample trajectories $\mathcal{D}_i = \{\tau\}, \tau \sim \pi_{\theta_i}$.
  Sample expert trajectories $\mathcal{D}_i^E \subset \mathcal{D}^E$.
  Update discriminator parameters from w_i to w_{i+1} with the gradient

$$\hat{\mathbb{E}}_{\mathcal{D}_i}[\nabla_w \log(D_w(s,a))] + \hat{\mathbb{E}}_{\mathcal{D}_i^E}[\nabla_w \log(1 - D_w(s,a))]$$

  Update the rewards in $\mathcal{D}_i$ with

$$r'(s,a) = r(a,b) - \lambda_1 \log(D_{w_i}(s,a)), \forall (s,a,r) \in \mathcal{D}_i$$

  Update the policy with policy gradient method (*e.g.*, TRPO, PPO) using the following gradient

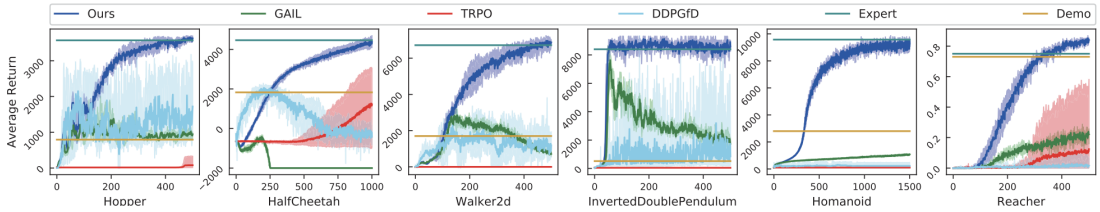
$$\hat{\mathbb{E}}_{\mathcal{D}_i}[\nabla_\theta \log \pi_\theta(a|s) Q'(s,a)] - \lambda_2 \nabla_\theta H(\pi_{\theta_i})$$

end for

4 Conclusion

- The potential future research directions:
  POfD is the first one that can acquire knowledge from few and imperfect demonstration data to aid exploration in environments with sparse feedback. I think the most powerful thing is that it can use few data to reach the dense reward performance in sparse reward. I think that its future research direction could transfer the reward reshaping mechanism into the hierarchical reinforcement learning. Such like with the early learning stage we use the dense reward to do the heuristic guided, and in later learning stage can use POFD in sparse reward to avoid the lack of reward knowledge problem.
- Any technical limitations:
  I think the limitation must be appear in the training process, since it use the concept of GAN to do the reward reshaping.If the learning of the generated network goes wrong, this. method wil be suffered the devastating performance descending.
- The results on the problem of interest:

Environment	$\mathcal{S}$	$\mathcal{A}$	Sparsification	Empirical Return		
				Demonstration	Expert	Ours
MountainCar-v1	$\mathbb{R}^2$	$\{0, 1, 2\}$	TYPE1	-165.0	-98.75	$\mathbf{-98.35 \pm 9.35}$
CartPole-v0	$\mathbb{R}^4$	$\{0, 1\}$	TYPE2	49	**500**	$\mathbf{500 \pm 0}$
Hopper-v1	$\mathbb{R}^{11}$	$\mathbb{R}^3$	TYPE3 (1 unit)	793.86	3571.38	$\mathbf{3652.23 \pm 263.62}$
HalfCheetah-v1	$\mathbb{R}^{17}$	$\mathbb{R}^6$	TYPE3 (15 unit)	1827.77	4463.46	$\mathbf{4771.15 \pm 646.96}$
Walker2d-v1	$\mathbb{R}^{17}$	$\mathbb{R}^6$	TYPE3 (1 unit)	1701.13	6717.08	$\mathbf{7687.47 \pm 394.97}$
DoublePendulum-v1	$\mathbb{R}^{11}$	$\mathbb{R}$	TYPE4	520.23	8399.86	$\mathbf{9116.08 \pm 1290.74}$
Humanoid-v1	$\mathbb{R}^{376}$	$\mathbb{R}^{17}$	TYPE3 (1 unit)	2800.05	9575.40	$\mathbf{9823.43 \pm 2015.48}$
Reacher-v1	$\mathbb{R}^{11}$	$\mathbb{R}^2$	TYPE1	0.73	0.75	$\mathbf{0.86 \pm 0.34}$



References