Paper Critique

Anirud N, CE21B014

Course: DA7400, Fall 2024, IITM
Paper: Discriminator-Weighted Offline Imitation Learning from Suboptimal Demonstrations
Date: 16 August 2024

Make sure your critique addresses the following points:

1. The problem the paper is trying to address
2. Key contributions of the paper
3. Proposed algorithm/framework
4. How the proposed algorithm addressed the described problem

Note: Be concise with your explanations. Unnecessary verbosity will be penalized. Please don't exceed 2 pages.

---

1 Problem Statement

Imitation Learning is about the agent learning to mimic the behavior policy used by the expert demonstration to generate the offline dataset. However, our offline dataset does not always have expert demonstrations. The presence of sub-optimal actions might lead to learning of poor policies in offline learning. This paper addresses problems when the dataset contains many sub-optimal actions, and we don't have access to the environment. Here, the agent has access to a small pre-collected dataset sampled from an expert and a large batch offline dataset sampled from one or multiple behavior policies that can be highly sub-optimal.

2 Key Contributions

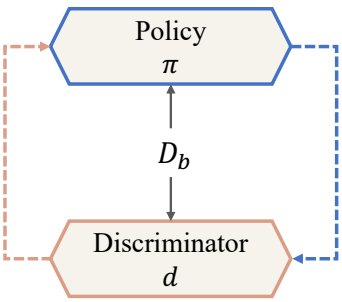
- A cooperative framework to learn the policy and the discriminator Discriminator is used to determine the quality of the demonstration in the dataset
- Effective and Light weighted offline IL with worst-case error minimization
- Comparison of the algorithm with the state-of-the-art methods
- The discriminator can perform offline policy selection.

3 Proposed Algorithm/Framework

3.1 Learning of Discriminator

$$\min_d \eta \mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}_e} [-\log d(s,a)] + \mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}_o} [-\log(1-d(s,a))] - \eta \mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}_e} [-\log(1-d(s,a))],$$

η is a hyperparameterter. and $d(s,a)$ is the discriminator. The discriminator is trained to output 1 for expert transitions and 0 for non-expert transitions. PU(positive unlabelled) learning - re-weight the losses for positive and unlabeled data. The first term in the equation is to encourage the network to give higher d values for expert transitions. The second term is to penalize when the actions from D_o are given high d values. The third term balances when the expert transitions and the transitions from D_o resemble one another.



New BC Task

$$L_\pi = \mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}_b}[-\log \pi(a|s) \cdot f(d(s,a,\log\pi))]$$

New Discriminating Task

$$L_d = \eta \mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}_e}[-\log d(s,a,\log\pi)] + \mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}_o}[-\log(1-d(s,a,\log\pi))] - \eta \mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}_e}[-\log(1-d(s,a,\log\pi))]$$

Figure 1: Framework

3.2 Cooperative Learning

The discriminator helps to generalize Behavior Cloning. The weight is applied to the total dataset D_b which is a union of D_0 and D_e. This framework aims to choose those transitions from D_0 which are likely to be expert transitions. If the policy π is optimal, assigning high probabilities to expert actions in expert states, the discriminator will benefit from an enhanced learning signal. It will become easier for the discriminator to distinguish between expert and non-expert transitions in D_o, as $\pi(a|s)$ will be large when (s,a) are derived from expert behaviors and small when they originate from non-expert behaviors.

3.3 Discrimitaor-Weighted Behaviour Learning

The paper has proved and shown that even if the discriminator initially misclassifies transitions, the BC weights are designed to mitigate this, ensuring that the policy still learns from D_e. As the discriminator improves, these weights become more accurate, leading to a better policy.

$$\min_\pi \alpha\mathbb{E}_{(s,a)\sim D_e}[-\log\pi(a|s)] - \mathbb{E}_{(s,a)\sim D_e}\left[-\log\pi(a|s)\cdot\frac{\eta}{d(1-d)}\right] + \mathbb{E}_{(s,a)\sim D_o}\left[-\log\pi(a|s)\cdot\frac{1}{1-d}\right]$$

Here, $\alpha \geq 1$ is a weight factor

$$\text{BC weights} = \begin{cases} \alpha - \frac{\eta}{d(1-d)}, & \text{if } (s,a)\in D_e \\ \frac{1}{1-d}, & \text{if } (s,a)\in D_o \end{cases}$$

The corrective loss L_w derived as:

$$L_w = \mathbb{E}_{(s,a)\sim D_e}\left[\log\pi(a|s)\cdot\left(\frac{\eta}{d}+\frac{\eta}{1-d}\right)\right] - \mathbb{E}_{(s,a)\sim D_o}\left[\log\pi(a|s)\cdot\frac{1}{1-d}\right]$$

Final Learning Objective :

$$\min_\pi \alpha\mathbb{E}_{(s,a)\sim D_e}[-\log\pi(a|s)] - L_w$$

4 Advantages and Conclusions of the Algorithm

- Effective use of the sub-optimal data and ease of implementation
- GANs suffer from instability while solving min max optimisation problems whereas this method gives a decoupled method of training π and d and computationally effecient
- In GANs, the generator and discriminator are in a competitive relationship. Generator produces data that the discriminator cannot differentiate between expert and non expert - leading to 0 sum game. The cooperative framework in this paper avoids such issues.