Paper Critique

Anirud N, CE21B014

Course: DA7400, Fall 2024, IITM
Paper: COMBO: Conservative Offline Model-Based Policy Optimization
Date: 14 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

Offline reinforcement learning (RL) suffers from the distributional shift between the offline dataset and the learned policy. This paper introduces a novel model-based offline RL algorithm that aims to address this issue by providing conservative estimates of the value function. The paper critiques the assumption of a **model error oracle**, which presupposes access to upper-bound estimates of model error. This assumption often fails in complex datasets and deep network scenarios, which this paper aims to rectify.

2 Key Contributions

The paper presents a model-based policy optimization method for offline RL, termed COMBO. This algorithm uses an actor-critic approach that learns from both offline data and synthetic data generated by the model. COMBO penalizes the value function in state-action pairs that are unsupported by the offline dataset, and it operates without assuming access to an uncertainty oracle. The paper also provides the following mathematical results:

- The Q-function of COMBO lower bounds the actual Q-function.
- The Q-function in COMBO is less conservative than model-free counterparts.
- The expected off-policy improvement in COMBO lower bounds the actual value.
- Safe improvement over behavior policy is demonstrated.

3 Proposed Algorithm/Framework

Algorithm 1 COMBO: Conservative Model-Based Offline Policy Optimization

Require: Offline dataset $\mathcal{D}$, rollout distribution $\mu(\cdot|)$, learned dynamics model $\widehat{T}_\theta$, initialized policy and critic π_ϕ and Q_ψ.

1: Train the probabilistic dynamics model $\widehat{T}_\theta(', r|,) = \mathcal{N}(\mu_\theta(,), \Sigma_\theta(,))$ on $\mathcal{D}$.
2: Initialize the replay buffer $\mathcal{D}_{\text{model}} \leftarrow \emptyset$.
3: **for** $i = 1, 2, 3, \cdots$ **do**
4: Collect model rollouts by sampling from μ and $\widehat{T}_\theta$, starting from states in $\mathcal{D}$. Add model rollouts to $\mathcal{D}_{\text{model}}$.
5: Conservatively evaluate π_ϕ^i by repeatedly solving eq.[1] to obtain $\widehat{Q}_\psi^{\pi_\phi^i}$ using samples from $\mathcal{D} \cup \mathcal{D}_{\text{model}}$.
6: Improve policy under the state marginal of d_f by solving eq[2] to obtain π_ϕ^{i+1}.
7: **end for**

The goal is to get a model based offline RL algorithm - that enables optimizing a lower bound on the policy performance, but without requiring uncertainty quantification.

3.1 Conservative Policy Evaluation

$$\hat{Q}^{k+1} \leftarrow \arg\min_{Q} \beta \left(\mathbb{E}_{s,a\sim\rho(s,a)}\left[Q(s,a)\right] - \mathbb{E}_{s,a\sim\mathcal{D}}\left[Q(s,a)\right]\right) + \frac{1}{2}\mathbb{E}_{s,a,s'\sim d_f}\left[\left(Q(s,a) - \mathcal{B}^\pi \hat{Q}^k(s,a)\right)^2\right]. \tag{1}$$

The idea is to penalize the Q function at states for which actions are not observed much in the dataset

3.2 Policy Improvement using a Conservative Critic

After learning the value function Q, policy improvement is done by :

$$\pi' \leftarrow \arg\max_{\pi} \mathbb{E}_{s\sim\rho, a\sim\pi(\cdot|s)}\left[\hat{Q}^\pi(s,a)\right] \tag{2}$$

4 Advantages and Conclusions of the Algorithm

- COMBO outperforms MOPO,MoRel(model-based), CQL(model-free).
- Better generalization results
- Less conservative than Model-free and more conservative than model-based algorithms
- Penalizes Q functions at state-action pairs that are not much observed in the dataset
- Backed up with a lot of theoretical proofs and guarantees
- This COMBO relies on the probabilistic dynamic model of the environment (Algorithm 1 step 2). Inaccuracies might lead to sub-optimal policies
- Computationally less efficient