Paper Critique

Anirud N, CE21B014

Course: DA7400, Fall 2024, IITM
Paper: Risk Averse Offline Reinforcement Learning
Date: 14 August 2024

---

1 Problem Statement

One of the major challenges in Offline RL is the Bootstrapping error. The dataset may not have much information about the values in the state action pair. The paper aims to implement the connection between risk aversion and distributional robustness. It aims to avoid actions that have high risk and those which are less frequent in the Dataset.

2 Key Contributions

The paper presents Offline Risk Averse Actor Critic algorithm - the Critic learns the full Value distribution and the risk averse actor - takes care of risk averse criteria. A variational Auto Encoder is used for Imitation Learning - reduces bootstrapping error

3 Proposed Algorithm/Framework

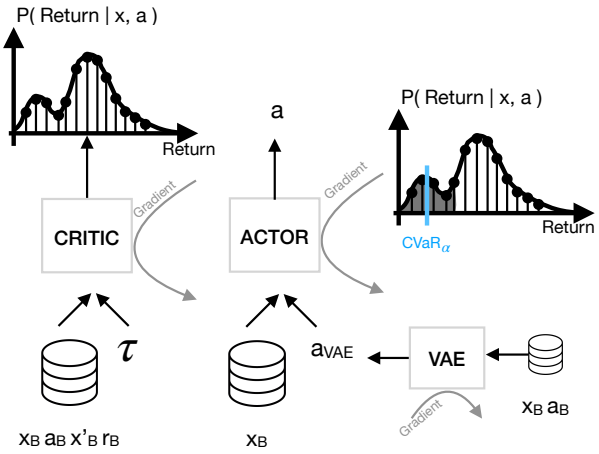

Figure 1: Framework

3.1 Distributional critic Learning

We introduce Distortion to the original Z (reward distribution) to account for risk.

$$Z^\pi(s,a) = \mathcal{D}\left(R(s,a) + \gamma Z^\pi(s',a')\right)$$

TD error :

$$\delta_{\tau,\tau'} = r + \gamma Z^\pi_{w'}(s',a';\tau') - Z^\pi_w(s,a;\tau), \tag{1}$$

Algorithm 1 Offline Risk-Averse Actor Critic (O-RAAC).

Require: Dataset, Critic Z_w and critic-target $Z_{w'}$, $\text{VAE}_\phi = \{E_{\phi_1}, D_{\phi_2}\}$, Perturbation model ξ_θ and target $\xi_{\theta'}$, modulation parameter λ, Distortion operator $\mathcal{D}$ or distortion sampling distribution $\mathcal{P}_\mathcal{D}$, All other hyperparameters

0: **for** $t = 1, \dots$ **do**
1: Sample B transitions (s, a, r, s') from the dataset.
2: Sample N quantiles τ and N' target quantiles τ' from $U(0, 1)$ and compute $\delta_{\tau,\tau'}$ in (2).
3: Compute policy $\pi_\theta = b + \lambda\xi_\theta(s, b)$, s.t. $b \sim \text{VAE}_\phi(s, a)$ as in (9).
4: Compute critic loss $\mathcal{L}_{\text{critic}}(w)$ in (4); actor loss $\mathcal{L}_{\text{actor}}(\theta)$ in (5); VAE loss $\mathcal{L}_{\text{VAE}}(\phi)$ in (10).
5: Gradient step $w \leftarrow w - \eta\nabla\mathcal{L}_{\text{critic}}(w)$; $\theta \leftarrow \theta - \eta\nabla\mathcal{L}_{\text{actor}}(\theta)$; $\phi \leftarrow \phi - \eta\nabla\mathcal{L}_{\text{VAE}}(\phi)$.
6: Perform soft-update on $w' \leftarrow \mu w + (1 - \mu)w'$; $\theta' \leftarrow \mu\theta + (1 - \mu)\theta'$.
6: **end for**=0

for τ, τ' independently sampled from the uniform distribution, i.e., $\tau, \tau' \sim \mathcal{U}(0, 1)$ and $a' \sim \pi(\cdot|s')$. The τ-quantile Huber-loss is

$$\mathcal{L}_\kappa(\delta; \tau) = \underbrace{\left|\tau - 1_{\{\delta<0\}}\right|}_{\text{Quantile loss}} \cdot \underbrace{\begin{cases} \frac{1}{2\kappa}\delta^2 & \text{if } |\delta| \le \kappa, \\ |\delta| - \frac{1}{2}\kappa & \text{otherwise.} \end{cases}}_{\text{Huber loss}} \tag{2}$$

The final critic loss is :

$$\mathcal{L}_{\text{critic}}(w) = \mathbb{E}_{\substack{(s,a,r,s')\sim d^\beta(\cdot) \\ a'\sim\pi(\cdot|s')}}\Big[\frac{1}{N \cdot N'}\sum_{i=1}^{N}\sum_{j=1}^{N'}\mathcal{L}_\kappa(\delta_{\tau_i,\tau'_j}; \tau_i)\Big]. \tag{3}$$

The critic uses a quantile function representation of the return distribution.

3.2 Risk Averse Actor

We prefer Deterministic Policies over stochastic ones. Actor Loss :

$$\mathcal{L}_{\text{actor}}(\theta) = -\mathbb{E}_{s\sim\rho^\beta(\cdot)}\left[\mathcal{D}\left(Z_w^{\pi_\theta}(s, \pi_\theta(s); \tau)\right)\right], \tag{4}$$

3.3 Off Policy to Online

$$\pi_\theta(s) = b + \lambda\xi_\theta(\cdot|s, b), \qquad \text{s.t., } b \sim \pi^{\text{IL}}(\cdot|s). \tag{5}$$

b is an action sampled from the imitation learning component, ξ_θ is a conditionally deterministic perturbation model that is optimized maximizing the actor loss Thus, all the randomness in our policy arises from the behaviour policy and not from the subsequent optimization. To generate $\pi^{\text{IL}}(\cdot|s)$ we use VAE.

$$\mu, \Sigma = E_{\phi_1}(s, a); \qquad z \sim \mathcal{N}(\mu, \Sigma); \qquad b = D_{\phi_2}(s, z), \tag{6}$$

where E_{ϕ_1} is the neural network encoder, D_{ϕ_2} is the neural network decoder

4 Advantages and Conclusions of the Algorithm

- Risk Averse algorithm
- Quantile function is used for effective computation
- VAEs do not suffer from mode collapse, ensuring a more stable training process.
- Aims to reduce bootstrap error by minimising the difference between behaviour policy and actor policy