Improving Robustness via Risk Averse Distributional Reinforcement Learning

Rahul Singh, Qinsheng Zhang, and Yongxin Chen

Abstract

- A risk-aware algorithm to learn robust policies for continuous control tasks within the distributional reinforcement learning (DRL) framework [1]
- Incorporate risk in sample based distributional policy gradient (SDPG) [3] for learning risk-averse policies to achieve robustness against system disturbances

Background

DRL: Instead of learning only Q-value, full distribution of returns is learned for each state-action pair

$$Q^\pi(x,a) = \mathbb{E}\, Z^\pi(x,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(x_t,a_t)\right], \quad x_t \sim P(\cdot \mid x_{t-1}, a_{t-1}), a_t = \pi(x_t)$$

- The return distribution satisfies distributional Bellman's equation

$$Z^\pi(x,a) = R(x,a) + \gamma Z^\pi(x', \pi(x') \mid x, a) \tag{1}$$

SDPG: Actor-critic type policy gradient method in DRL representing the return distribution by *samples* via reparametrization

- The critic network G_ϕ mimics the return distribution determined via distributional Bellman equation based on samples
  - Uses quantile Huber loss:

$$L_{critic}(\phi) = \mathbb{E}\left[\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} \rho_{\tau_i}^{\zeta}(\tilde{z}_j - z_i)\right] \tag{2}$$

- The actor network π_θ parameterizes the policy

$$\nabla_\theta L_{actor}(\theta) = \mathbb{E}\left[\nabla_\theta \pi_\theta(x)\, \frac{1}{n}\sum_{j=1}^{n}\left[\nabla_a z_j\right]|_{a=\pi_\theta(x)}\right] \tag{3}$$

Risk Measures: Exponential utility function, cumulative probability weighting, distortion risk measures, VaR, CVaR, etc.

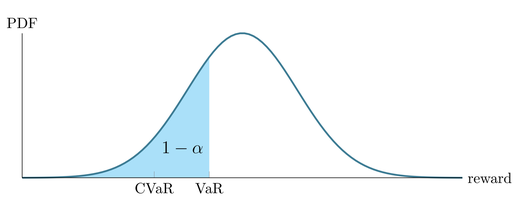
- Conditional value at risk (CVaR) [2] of a random variable Z at level $\alpha \in [0,1]$

$$\mathrm{CVaR}_\alpha(Z) := \mathbb{E}\left[Z \mid Z \le \mathrm{VaR}_\alpha(Z)\right] \tag{4}$$

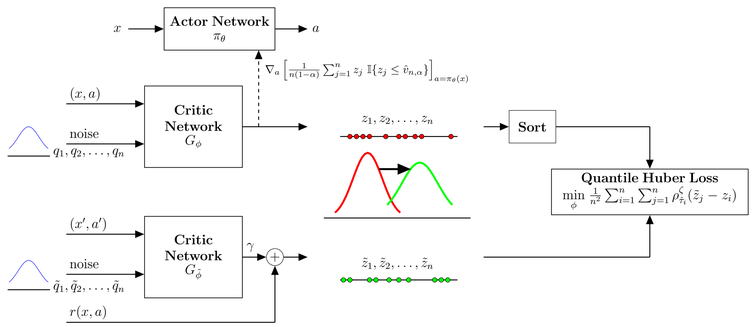
- Let $\{z_i\}_{i=1}^n$ be the i.i.d. samples from the distribution of Z and let $\{z_{[i]}\}_{i=1}^n$ be its order statistics with $z_{[1]} \le z_{[2]} \le \ldots \le z_{[n]}$. Then CVaR at level α

$$\hat{c}_{n,\alpha} = \frac{1}{n(1-\alpha)}\sum_{i=1}^{n} z_i\, \mathbb{I}\{z_i \le \hat{v}_{n,\alpha}\}, \tag{5}$$

where $\hat{v}_{n,\alpha}$ is estimated VaR from samples $\hat{v}_{n,\alpha} = z_{\lfloor n(1-\alpha)\rfloor}$



Approach



- Utilize the return distribution learned via SDPG to incorporate risk
- CVaR as the risk-measure to learn the policy
- Risk-sensitive SDPG
  - Two networks: critic network G_ϕ and actor network π_θ
  - Gradient of the actor network loss function

$$\nabla_\theta L_{actor}(\theta) = \mathbb{E}\left[\nabla_\theta \pi_\theta(x)\, \nabla_a \left[\frac{1}{n(1-\alpha)}\sum_{j=1}^{n} z_j\, \mathbb{I}\{z_j \le \hat{v}_{n,\alpha}\}\right]_{a=\pi_\theta(x)}\right] \tag{6}$$

Algorithm 1 Risk-averse SDPG

Require: Learning rates β_1 and β_2, CVaR level α, batch size M, sample size n
Initialize the the actor network (π) parameters θ, critic network (G) parameters ϕ
Initialize target networks $(\tilde{\theta}, \tilde{\phi}) \leftarrow (\theta, \phi)$
for the number of environment steps **do**
  Sample M number of transitions $\{(x_t^i, a_t^i, r_t^i, x_{t+1}^i)\}_{i=1}^M$ from the replay pool
  Sample noise $\{q_j^i\}_{j=1}^n \sim \mathcal{N}(0,1)$ and $\{\tilde{q_j}^i\}_{j=1}^n \sim \mathcal{N}(0,1)$, for $i = 1, \ldots, M$
  Apply Bellman update to create samples (of return distribution)

$$\tilde{z}_j^i = r_t^i + \gamma G_{\tilde{\phi}}(\tilde{q_j}^i | (x_{t+1}^i, \pi_{\tilde{\theta}}(x_{t+1}^i))) \quad \text{for } j = 1, 2, \ldots, n$$

  Generate samples $z_j^i = G_\phi(q_j^i | (x_t^i, a_t^i)) \quad \text{for } j = 1, 2, \ldots, n$
  Sort the samples z^i in ascending order
  Update G_ϕ by stochastic gradient descent with learning rate β_1:

$$\frac{1}{M}\sum_{i=1}^{M}\frac{1}{n^2}\sum_{j=1}^{n}\sum_{k=1}^{n}\rho_{\tau_j}^{\zeta}(\tilde{z}_k^i - z_j^i)$$

  Update π_θ by stochastic gradient ascent with learning rate β_2:

$$\frac{1}{M}\sum_{i=1}^{M}\pi_\theta(x_t^i)\, \nabla_a \left[\frac{1}{n(1-\alpha)}\sum_{j=1}^{n} z_j^i\, \mathbb{I}\{z_j^i \le z^i_{\lfloor n(1-\alpha)\rfloor}\}\right]_{a=\pi_\theta(x_t^i)}$$

  Update target networks $(\tilde{\theta}, \tilde{\phi}) \leftarrow (\theta, \phi)$
end for

Results

- Robustness of risk-averse SDPG algorithm against system disturbances on multiple OpenAI Gym continuous control tasks
- Evaluating the robustness of learned policies: different levels of disturbances on action forces to at different α levels of CVaR

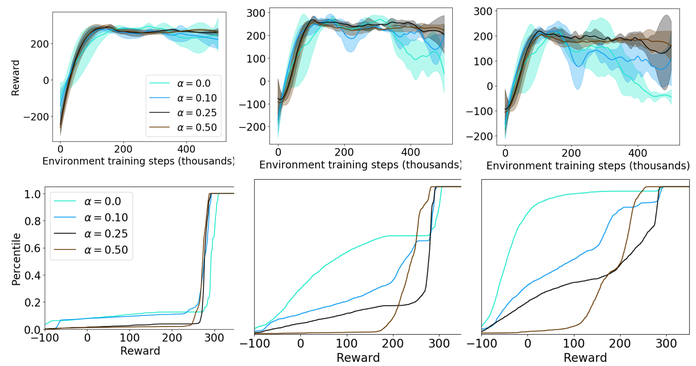


Figure 1: *BipedalWalker-v2. Top row depicts evaluation curves and the bottom row depicts the CDFs at different noise levels. The evaluations are done every 5000 environment steps in each trial over 1000 episodes. The shaded region represents standard deviation of the average returns over 5 random seeds. The first column is noise-free, NoiseLevel = 0. The second column is corresponding to NoiseLevel = 1.0. The final column is NoiseLevel = 1.5.*

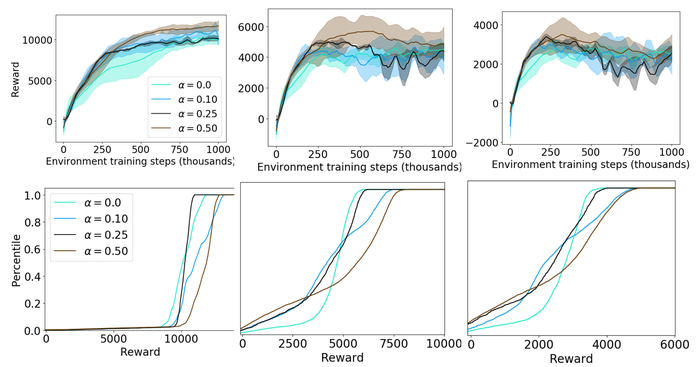


Figure 2: *HalfCheetah-v2*

References

[1] Marc G Bellemare, Will Dabney, and Rémi Munos. "A distributional perspective on reinforcement learning". In: *Proceedings of the 34th International Conference on Machine Learning*. Vol. 70. 2017, pp. 449–458.

[2] Yinlam Chow and Mohammad Ghavamzadeh. "Algorithms for CVaR optimization in MDPs". In: *Advances in neural information processing systems*. 2014, pp. 3509–3517.

[3] Rahul Singh, Keuntaek Lee, and Yongxin Chen. "Sample-based Distributional Policy Gradient". In: *arXiv preprint arXiv:2001.02652* (2020).