RISK-AVERSE OFFLINE REINFORCEMENT LEARNING

Presented by Jia Chengxing

LAMDA, Nanjing University

February 19, 2021

南京大学 NANJING UNIVERSITY

LAMDA Learning And Mining from DatA

Table of Contents

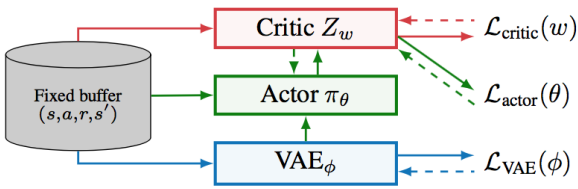
Table of Contents

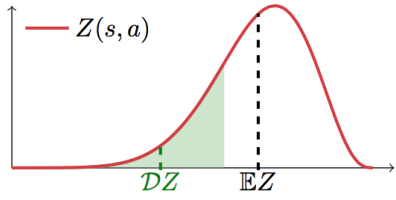
Background and Contribution

- Although several algorithms for risk-sensitive RL exist, none of them addresses the offline setting.
- On the other hand, existing offline RL algorithms consider the average performance criterion and are risk-neutral.
- Present the first approach towards learning a risk-averse RL policy for high-stakes applications using only offline data: the Offline Risk-Averse Actor-Critic (O-RAAC).

Background and Contribution

Three components: a distributional critic that learns the full value distribution, a risk-averse actor that optimizes a risk averse criteria and an imitation learner implemented with a variational auto-encoder (VAE) that reduces the bootstrapping error due to the offline nature of the algorithm.





Risk-averse RL

In risk-neutral RL, the goal is to find a policy that maximizes the expected discounted sum of returns $\mathbb{E}_{d_\pi}[\sum_{t=0}^{\infty} \gamma^t R(\cdot|s,a)]$.
In risk-averse settings, the goal is to find the policy π that maximizes $\mathcal{D}[\sum_{t=0}^{\infty} \gamma^t R(\cdot|s,a)]$.

- Conditional Value-at-Risk (CVaR): Using the distributional of value, recent work used a Gaussian distribution.
- Cumulative Prospect Theory or Exponential Utility.

Offline RL

The biggest challenge in offline RL is the Bootstrapping Error: a Q-function is evaluated at state-action pairs where there is little or no data and these get propagated through the Bellman equation. In turn, a policy optimized with offline data induces a state-action distribution that is shifted from the original data.

- Model-free: express the actor as the sum between an imitation learning component and a perturbation model, regularizing the policies with the behavior policy using the MMD distance or f-divergences.
- Model-based methods with pessimistic MDP.

Distributional RL

Considering the distribution of value instead of mean of value.

- Categorical representation: C51.
- Quantile representation: QR-DQN.
- Implicit Quantile Network (IQN).

Table of Contents

Distributional Critic

To learn the distributional critic, exploit the distributional Bellman equation of returns $Z^\pi(s,a) =_{\mathcal{D}} R(s,a) + \gamma Z^\pi(S', A')$, the random variable S', A' are distributed according to $s' \sim p(\cdot|s,a)$ and $a' \sim \pi(\cdot|s')$.

Use a target network ω' and compute the temporal difference (TD) error at a sample (s, a, s', r) as

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

with τ, τ' independently sampled from the uniform distribution.

The τ-huber loss:

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

and the critic loss:

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

Risk-averse Actor

prefer deterministic policies over stochastic ones because introducing extra randomness is against a risk-averse behavior.
Consider parameterized deterministic policies $\pi_\theta(s)$. Define the actor loss as:

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

there exists a quantile sampling distribution $\mathbb{P}_{\mathcal{D}}$

$$\mathcal{D}\left(Z_w^{\pi_\theta}(s,\pi_\theta(s);\tau)\right) = \int Z_w^{\pi_\theta}(s,\pi_\theta(s);\tau)\mathbb{P}_{\mathcal{D}}(\tau)\,\mathrm{d}\tau \approx \frac{1}{K}\sum_{k=1}^{K} Z_w^{\pi_\theta}(s,\pi_\theta(s);\tau_k),\ \tau_k \sim \mathbb{P}_{\mathcal{D}}$$

and the CVaR:

$$\mathbf{CVaR}_\alpha(Z_w^{\pi_\theta}(s,a;\tau)) = \frac{1}{\alpha}\int_0^\alpha Z_w^{\pi_\theta}(s,a;\tau)\,\mathrm{d}\tau$$

Off-policy to Offline

Like BCQ, express the actor as the sum between an imitation learning component and a perturbation model:

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

and the VAE loss:

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

$$\mathcal{L}_{\text{VAE}}(\phi) = \mathbb{E}_{s,a\sim\beta(\cdot)} \left[\underbrace{(a - D_{\phi_2}(s, z))^2}_{\text{reconstruction loss}} + \frac{1}{2} \underbrace{\text{KL}(\mathcal{N}(\mu, \Sigma), \mathcal{N}(0, I))}_{\text{regularization}} \right]$$

O-RAAC

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

input Data set, 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 $\mathbb{P}_\mathcal{D}$, critic-loss parameters N, N', κ, mini-batch size B, learning rate η, soft update parameter μ.

for $t = 1, \ldots$ **do**

Sample B transitions (s, a, r, s') from data set.

Sample N quantiles τ and N' target quantiles τ' from $\mathcal{U}(0, 1)$ and compute $\delta_{\tau,\tau'}$ in (2).

Compute policy $\pi_\theta = b + \lambda\xi_\theta(s, b)$, s.t. $b \sim \text{VAE}_\phi(s, a)$ as in (9).

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).

Gradient step $w \leftarrow w - \eta\nabla\mathcal{L}_{\text{critic}}(w)$; $\theta \leftarrow \theta - \eta\nabla\mathcal{L}_{\text{actor}}(\theta)$; $\phi \leftarrow \phi - \eta\nabla\mathcal{L}_{\text{VAE}}(\phi)$.

Perform soft-update on $w' \leftarrow \mu w + (1 - \mu)w'$; $\theta' \leftarrow \mu\theta + (1 - \mu)\theta'$.

end for

Table of Contents

RAAC in off-policy

As a toy example, we chose a 1-D car with state $s = (x, v)$, for position and velocity. The agent controls the car with an acceleration $a \in [-1, 1]$. The car dynamics with a time step $\Delta t = 0.1$ is

$$x_{t+1} = x_t + v_t \Delta t + 0.5 a_t (\Delta t)^2, \qquad v_{t+1} = v_t + a_t \Delta t.$$

The control objective is to move the car to $x_{\mathrm{g}} = 2.5$ as fast as possible, starting from rest. To model the risk of crashing or of getting a speed fine, we introduce a penalization when the car exceeds a speed limit ($v > 1$). Hence, we use a random reward function given by

$$R_t(s, a) = -10 + 370 \mathbb{I}_{x_t = x_{\mathrm{g}}} - 25 \mathbb{I}_{v_t > 1} \cdot \mathcal{B}_{0.2},$$

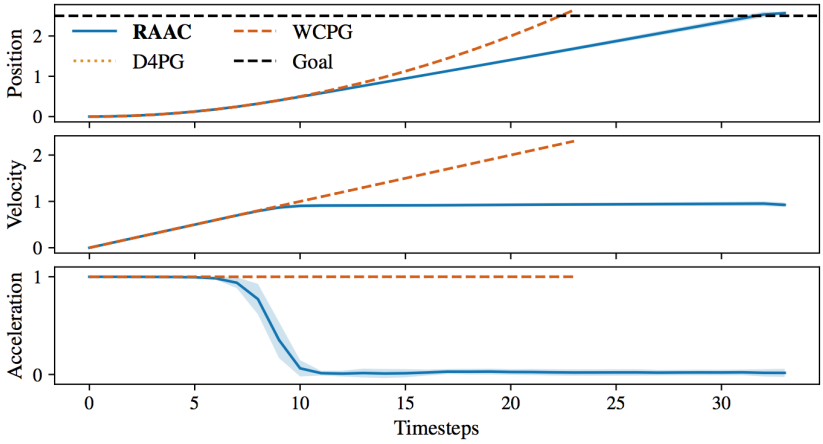
where $\mathbb{I}$ is an indicator function and $\mathcal{B}_{0.2}$ is a Bernoulli Random Variable with probability $p = 0.2$. The episode terminates after 400 steps or when the agent reaches the goal.

RAAC in off-policy

Table 1: Results of RAAC, WCPG, and D4PG in the car example. RAAC learns a policy that saturates the velocity before the risky region. WCPG and D4PG learn to accelerate as fast as possible, reaching the goal first with highest average returns but suffer from events with large penalty. We report mean (standard deviation) of each quantity.

Algorithm	$\text{CVaR}_{0.1}$	Mean	Risky Steps	Total Steps
RAAC	**48.0 (8.3)**	48.0 (8.3)	**0 (0)**	33 (1)
WCPG	15.8 (3.3)	**79.8 (1.3)**	13 (0)	**24 (0)**
D4PG	15.6 (4.4)	**79.8 (2.0)**	13 (0)	**24 (0)**

RAAC in off-policy

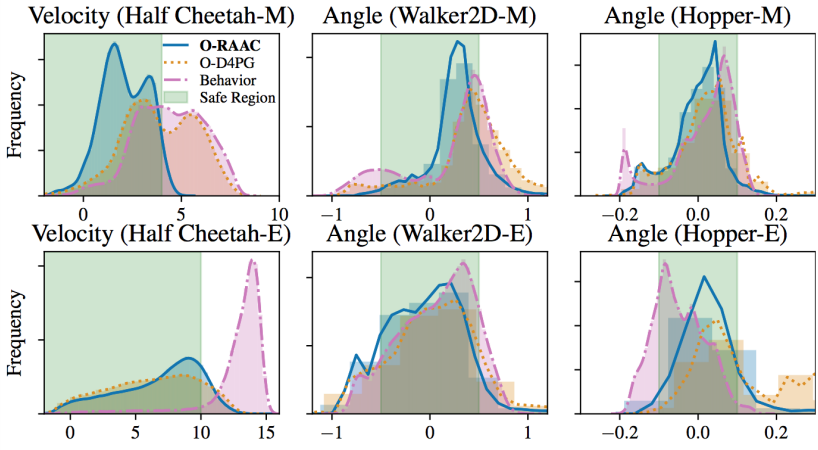


Risk-averse Offline

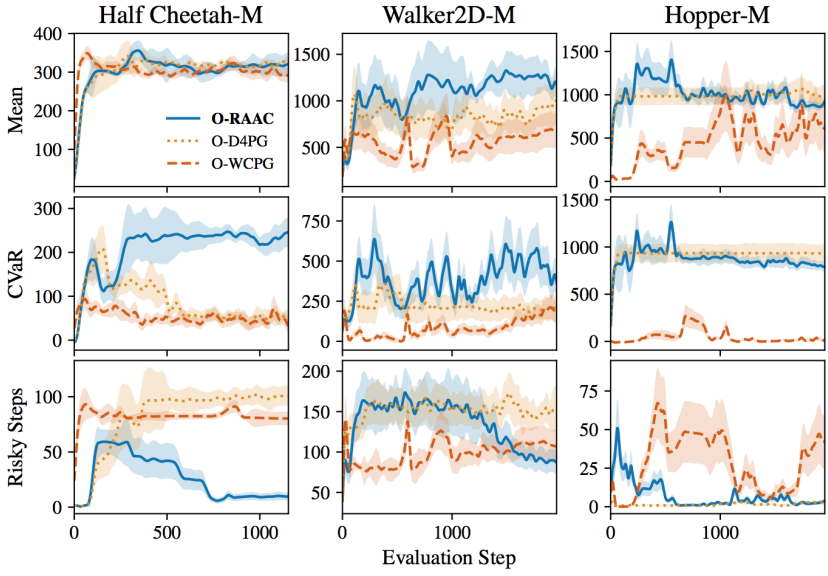
Half-Cheetah: $R_t(s,a) = \bar{r}_t(s,a) - 70\mathbb{I}_{v>\bar{v}} \cdot \mathcal{B}_{0.1}$, where $\bar{r}_t(s,a)$ is the original environment reward, v the forward velocity, and $\bar{v}$ is a threshold velocity ($\bar{v} = 4$ for the (M) variant and $\bar{v} = 10$ for the (E) variant). As with the car example, this high-velocity penalization models a penalty to the rare but catastrophic event of the robot breaking – we want to be risk-averse to it. We evaluate the Half-Cheetah for 200 time steps.

Walker2D/Hopper: $R_t(s,a) = \bar{r}_t(s,a) - p\mathbb{I}_{|\theta|>\bar{\theta}} \cdot \mathcal{B}_{0.1}$, where $\bar{r}_t(s,a)$ is the original environment reward, θ is the pitch angle, $\bar{\theta}$ is a threshold angle ($\bar{\theta} = 0.5$ for the Walker2d-M/E and $\bar{\theta} = 0.1$ for the Hopper-M/E) and $p = 30$ for the Walker2d-M/E and $p = 50$ for the Hopper-M/E. When $|\theta| > 2\bar{\theta}$ the robot falls, the episode terminates, and we stop collecting such rewards. To avoid such situation, we shape the rewards with the stochastic event at $\theta > \bar{\theta}$. The maximum duration of the Walker2D and the Hopper is 500 time steps.

Risk-averse Offline



Risk-averse Offline



Risk-neutral Offline

$$\max_{\pi} \mathcal{D}\left[Z^{\pi}(x,a)\right] = \max_{\pi} \min_{d \in \mathcal{D}_{\pi}} \mathbb{E}_{d}\left[Z^{\pi}(x,a)\right]$$

	Algorithm	Medium CVaR$_{0.1}$	Medium Mean	Medium Duration	Expert CVaR$_{0.1}$	Expert Mean	Expert Duration
Half-Cheetah	**O-RAAC$_{0.1}$**	**214 (36)**	**331 (30)**	200 (0)	**595 (191)**	**1180 (78)**	200 (0)
	O-RAAC$_{0.25}$	**252 (14)**	**317 (5)**	200 (0)	**695 (34)**	**1185 (7)**	200 (0)
	O-RAAC$_{\text{CPW}}$	**253 (9)**	**318 (3)**	200 (0)	358 (67)	974 (21)	200 (0)
	O-WCPG	76 (14)	**316 (23)**	200 (0)	248 (232)	**905 (107)**	200 (0)
	O-D4PG	66 (34)	**341 (20)**	200 (0)	**556 (263)**	**1010 (153)**	200 (0)
	BEAR	15 (30)	**312 (20)**	200 (0)	44 (20)	557 (15)	200 (0)
	RAAC	-55 (1)	-52 (0)	200 (0)	3 (13)	30 (3)	200 (0)
	VAE	10 (23)	**354 (9)**	200 (0)	260 (84)	754 (18)	200 (0)
	Behavior	9 (6)	**344 (2)**	200 (0)	100 (8)	727 (4)	200 (0)
Walker-2D	**O-RAAC$_{0.1}$**	**751 (154)**	**1282 (20)**	397 (18)	**1172 (71)**	**2006 (56)**	**432 (11)**
	O-RAAC$_{0.25}$	497 (71)	**1257 (27)**	**479 (6)**	670 (133)	1758 (48)	**436 (7)**
	O-RAAC$_{\text{CPW}}$	500 (71)	**1304 (16)**	**477 (3)**	819 (89)	1874 (34)	**454 (8)**
	O-WCPG	-15 (41)	283 (37)	185 (12)	362 (33)	1372 (160)	301 (31)
	O-D4PG	31 (29)	308 (20)	249 (9)	773 (55)	1870 (63)	405 (12)
	BEAR	517 (66)	**1318 (31)**	**468 (8)**	1017 (49)	1783 (32)	**463 (4)**
	RAAC	55 (2)	92 (9)	200 (7)	54 (2)	83 (6)	196 (6)
	VAE	-84 (21)	425 (37)	246 (9)	345 (302)	1217 (180)	**350 (130)**
	Behavior	-56 (9)	727 (16)	500 (0)	1028 (34)	1894 (7)	500 (0)
Hopper	**O-RAAC$_{0.1}$**	**1416 (28)**	1482 (4)	**499 (1)**	**980 (28)**	**1385 (33)**	**494 (6)**
	O-RAAC$_{0.25}$	1108 (14)	1337 (21)	419 (6)	**730 (129)**	1304 (21)	434 (6)
	O-RAAC$_{\text{CPW}}$	969 (9)	1188 (6)	373 (2)	488 (1)	496 (0)	160 (0)
	O-WCPG	-87 (25)	69 (8)	100 (0)	720 (34)	898 (12)	301 (1)
	O-D4PG	1008 (28)	1098 (11)	359 (3)	606 (31)	783 (18)	268 (3)
	BEAR	1252 (47)	**1575 (8)**	481 (2)	852 (30)	1180 (12)	431 (4)
	RAAC	71 (23)	113 (5)	146 (4)	474 (0)	475 (0)	**500 (0)**
	VAE	727 (39)	1081 (17)	462 (4)	774 (36)	1116 (13)	**498 (1)**
	Behavior	674 (5)	1068 (4)	500 (0)	827 (12)	1211 (3)	500 (0)

Table of Contents

Conclusion

DPG+BCQ+IQN...

Reference