Risk-averse Offline Reinforcement Learning

Núria Armengol-Urpí, Sebastian Curi, Andreas Krause

ETH zürich

Motivation

- In high stakes applications, we would like to do well even in **rare** events.
- Risk-averse RL focuses on large-but-rare losses and assigns more weight to adverse events rather than to positive ones.
- Deploying of existing risk-averse RL agents in safety-crucial applications is limited by catastrophic events occurring at early exploration stages.
- Offline RL setting considers learning a policy only from fixed pre-collected data.
- None of the existing offline RL algorithms are risk-averse but risk-neutral, i.e., may sacrifice large-but-rare losses for the sake of performing well in average.

Related Work

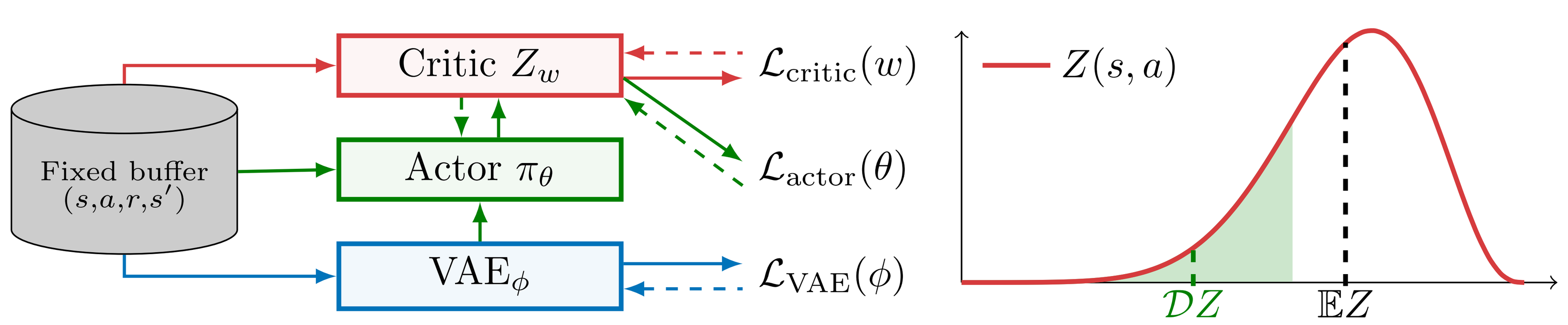
- Most of the previous work in risk-averse RL require known tabular MDPs (Chow et al., 2015) or are limited to the on-policy setting (Tamar et al., 2015).
- Existing offline RL algorithms (Fujimoto et al. 2019, Kumar et al., 2019) are risk-neutral.

O-RAAC

O-RAAC is an approach for learning a risk-averse RL policy using offline data.

The algorithm has 3 components:

- **Critic:** Distributional component that learns the full value distribution.
- **Actor:** Risk-averse component that optimizes a desired risk-averse criteria.
- **VAE:** Imitation learner that reduces the bootstrapping error.



- Data collected by a pre-trained agent is stored in a buffer for offline training.
- The VAE learns a generative model of the behavior policy.
- The actor is a deterministic perturbation model that perturbs the VAE in the direction of maximizing a risk-averse distortion $\mathcal{D}$ of the Z-value distribution.
- The Z-value distribution of the policy is learnt by the critic.

Distributional critic: We represent the Z-value distribution through its quantile function as proposed by Dabney et al., (2018) but extend it to the continuous setting. We parameterize it through a NN with learnable parameters w: $Z_w(s,a;\tau)$.

Risk-averse actor: We can approximate a risk distortion $\mathcal{D}$ of Z via sampling from an associated quantile distribution $\mathbb{P}_{\mathcal{D}}$:

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

and optimize the risk-averse actor to maximize (to be less risky) such quantity.

Online to offline: The policy π_θ uses a similar parameterization than in Fujimoto et al. (2019) and can be decomposed as:

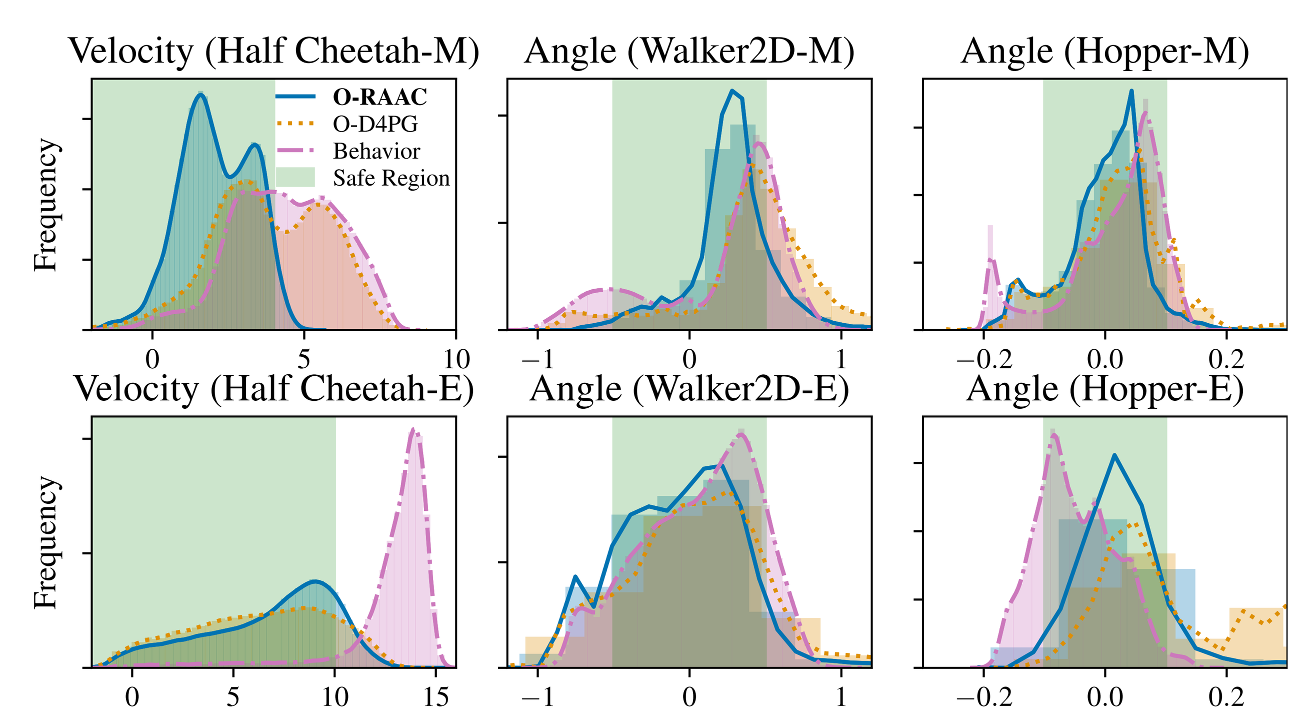
$$\pi_\theta(s) = \underbrace{b}_{\text{Imitation learning component for bootstrapping error reduction}} + \underbrace{\overbrace{\lambda}^{\text{Hyperparameter scaling perturbation magnitude}}\xi_\theta(\cdot|s,b)}_{\text{Reinforcement Learning component for risk-aversity}}, \quad \text{s.t.}, b \sim \mathrm{VAE}_\phi(\cdot|s).$$

Experimental Results

We test performance of O-RAAC on the D4RL dataset (Fu et al., 2020). We use 3 MuJoCo tasks: HalfCheetah, Walker2D and Hopper with medium (M) and expert (E) variants for each.
We introduce large-but-rare penalizations on the original deterministic reward function to model risk of agents falling or crashing when exceeding a speed limit (HalfCheetah) or a pitch angle threshold (Walker2D and Hopper).
We optimize the risk distortion $\mathcal{D} = \mathrm{CVaR}_{0.1}$.

Qualitative evaluation:

We show support of risk-events for O-RAAC, O-D4PG (risk neutral algorithm) and the dataset.



- O-RAAC learns to shift the support to the risk-free region (green area).
- O-D4PG ignores the rare penalties in the risky region and imitates the behavior policy distribution by having most of the support in the risky region.

Quantitative evaluation:

We compare O-RAAC, with other benchmarks in terms of risk-averse performance (“$\mathrm{CVaR}_{0.1}$” column) and risk-neutral performance (“Mean” column).

	Algorithm	Medium $\mathrm{CVaR}_{0.1}$	Medium Mean	Medium Duration	Expert $\mathrm{CVaR}_{0.1}$	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$_{\mathrm{CPW}}$	**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$_{\mathrm{CPW}}$	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$_{\mathrm{CPW}}$	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)

- O-RAAC has higher $\mathrm{CVaR}_{0.1}$ than all benchmarks.
- O-RAAC is compatible with different risk-averse criteria.
- In environments that terminate, O-RAAC has longer duration than competitors.
- O-RAAC has better or similar risk-neutral performance than benchmarks.
- Optimizing a risk-averse performance is beneficial to maximize the risk-neutral performance due to the distributional-robust properties of risk-sensitive criteria.

References

Yinlam Chow, Aviv Tamar, Shie Mannor, and Marco Pavone. Risk-sensitive and robust decision making: A CVaR optimization approach. *Advances in Neural Information Processing Systems, 2015.*

Aviv Tamar, Yonatan Glassner, and Shie Mannor. Optimizing the CVaR via sampling. *Proceedings of the National Conference on Artificial Intelligence, 4:2993–2999, 2015.*

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. *In 36th International Conference on Machine Learning, ICML 2019, 2019.*

Aviral Kumar, Justin Fu, George Tucker, and Sergey Levine. Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction. *NeurIPS, 2019*

Will Dabney, Georg Ostrovski, David Silver, and Remi Munos. Implicit quantile networks for distributional reinforcement learning. *In 35th International Conference on Machine Learning, ICML 2018, 2018.*

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: Datasets for Deep Data-Driven Reinforcement Learning, 2020.