Robust Reinforcement Learning with Distributional Risk-averse formulation

P. Clavier[1,2], **S. Allassonnière**[2], **E. Le Pennec**[1]

[1]CMAP, Ecole polytechnique, France,

[2]INRIA, INSERM, Université Paris Cité

pierre.clavier@polytechnique.edu

Abstract

Robust Reinforcement Learning tries to make predictions more robust to changes in the dynamics or rewards of the system. This problem is particularly important when the dynamics and rewards of the environment are estimated from the data. In this paper, we approximate the Robust Reinforcement Learning constrained with a Φ-divergence using an approximate Risk-Averse formulation. We show that the classical Reinforcement Learning formulation can be robustified using standard deviation penalization of the objective. Two algorithms based on Distributional Reinforcement Learning, one for discrete and one for continuous action spaces are proposed and tested in a classical Gym environment to demonstrate the robustness of the algorithms.

1 Introduction

Considering a Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, P, \gamma)$ with:

- $\mathcal{S}$ is the state space,
- $\mathcal{A}$ is the action space,
- $P(s', r \mid s, a)$ is the reward and transition distribution from state s to s' taking action a,
- $\gamma \in (0, 1)$ is the discount factor.

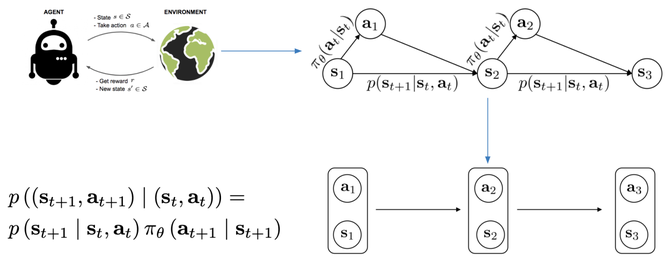


Figure 1: Reinforcement Learning as a MDP.

- Find the best stochastic policy are denoted $\pi(a \mid s) : \mathcal{S} \to \Delta(\mathcal{A})$.
- Distribution of discounted cumulative return : $Z^{P,\pi}(s,a) = R(\tau)$ with $R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t$.
- The Q-function $Q^{P,\pi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ of π s.t $Q^{P,\pi}(s,a) := \mathbb{E}[Z^{P,\pi}(s,a)]$.
- The Bellman operator $\mathcal{T}^\pi$ and Bellman optimal operator $\mathcal{T}^*$ are defined as follow:
  - $\mathcal{T}^\pi Q(s,a) := r(s,a) + \gamma \mathbb{E}_{P,\pi}\left[Q\left(s', a'\right)\right]$
  - $\mathcal{T}^* Q(s,a) := r(s,a) + \gamma \mathbb{E}_{P}\left[\max_{a'} Q\left(s', a'\right)\right]$.

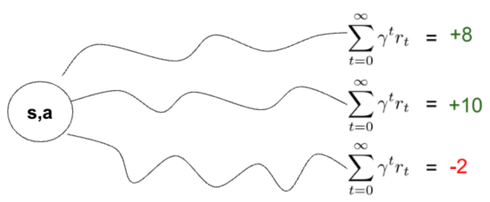


Figure 2: Some realisation of the sum of returns given (s,a)

2 Reinforcement Learning as an AMPI

- Reinforcement Learning can be viewed as policy iteration that can be viewed an an *Approximate Modified Policy Iteration: [4]*

$$\begin{cases} \pi_{k+1} \in \mathcal{G}\left(Q_k\right) = \arg\max_{\pi \in \Pi} \langle Q_k, \pi \rangle \Leftarrow \textbf{Improvement} \\ Q_{k+1} = \left(T^{\pi_{k+1}}\right)^m Q_k + \epsilon_{k+1} \Leftarrow \textbf{Evaluation} \end{cases}$$

with $\mathcal{T}^\pi Q(s,a) := r(s,a) + \gamma \mathbb{E}_{P,\pi}\left[Q\left(s', a'\right)\right]$ the Bellman Operator.

- Robust Reinforcement Learning problem :

$$\begin{cases} \pi_{k+1} \in \mathcal{G}'\left(Q_k\right) = \arg\max_{\pi \in \Pi} \langle \min_P Q_k^{(P,\pi)}, \pi \rangle \\ Q_{k+1} = \left(T^{\pi_{k+1}}\right)^m Q_k + \epsilon_{k+1} \end{cases}$$

3 Robust formulation in greedy step of AMPI.

- **Idea :** replace the step $\pi' \in \mathcal{G}(Q) = \arg\max_{\pi \in \Pi} \langle \min_P Q^{(P,\pi)}, \pi \rangle$ by a simpler and tractable expression.
- **Result :**

$$\underbrace{\min_{P \in D_{\chi^2}(P \| P_0) \leq \alpha} Q^{(P,\pi)}}_{\textit{Robust formulation}} = \underbrace{Q^{(P_0,\pi)} - \alpha \mathbb{V}[Z^{P_0}]^{\frac{1}{2}}}_{\textit{Risk-averse formulation}}$$

with :

- Z the **distribution of returns** : $Q^{P,\pi}(s,a) := \mathbb{E}[Z^{P,\pi}(s,a)]$.
- Condition on $\alpha \leq \frac{\mathbb{V}[Z^{P_0}]}{\left\|\tilde{Z}^{P_0}\right\|_\infty^2} \leq 1$ with $\tilde{Z}^{P_0} = Z^{P_0} - \mathbb{E}[Z^{P_0}]$.
- Q and $V[Z]$ can be estimated using Distributional Reinforcement Learning.

So we defined

$$\pi' \in \mathcal{G}_\alpha(Q) = \arg\max_{\pi \in \Pi} \langle Q^{(P_0,\pi)} - \alpha \mathbb{V}_{P_0}[Z]^{\frac{1}{2}}, \pi \rangle$$

and now look at the current AMPI to improve robustness :

$$\begin{cases} \pi_{k+1} \in \mathcal{G}_\alpha\left(Q_k\right) \\ Q_{k+1} = \left(T^{\pi_{k+1}}\right)^m Q_k + \epsilon_{k+1} \end{cases}$$

4 Distributional RL

- Distributional RL estimates $Z^\pi(s,a)$ the entire distribution.
- Usually in practice we represent Z as with $Z_\psi(s,a) := \frac{1}{M}\sum_{m=1}^{M} \delta\left(\theta_\psi^m(s,a)\right)$, a mixture of atoms-Dirac delta functions
- Parameters ψ of a neural network are obtained by minimizing the average over the 1-Wasserstein distance between Z_ψ and the temporal difference target distribution $\mathcal{T}^\pi Z_\psi$: [1].

$$\mathcal{T}^\pi Z(s,a) = \mathcal{R}(s,a) + \gamma Z\left(s', a'\right)$$
$$\text{with } s' \sim \mathcal{P}(\cdot \mid s,a), a' \sim \pi\left(\cdot \mid s'\right).$$

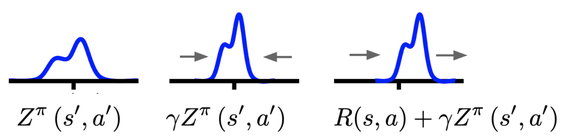


Figure 3: Illustration of Distributional Bellman Operator

- According to [1], the minimization of the 1-Wasserstein loss can be done by learning quantile locations for fractions $\tau_m = \frac{2m-1}{2M}, m \in [1..M]$ via quantile regression loss defined as :

$$\mathcal{L}_{\mathrm{QR}}^\tau(\theta) := \mathbb{E}_{\tilde{Z} \sim Z}\left[\rho_\tau(\tilde{Z} - \theta)\right], \text{ with}$$
$$\rho_\tau(u) = u(\tau - \mathbb{I}(u < 0)), \forall u \in \mathbb{R}.$$

- The quantile representation has the advantage of not fixing the support of the learned distribution.

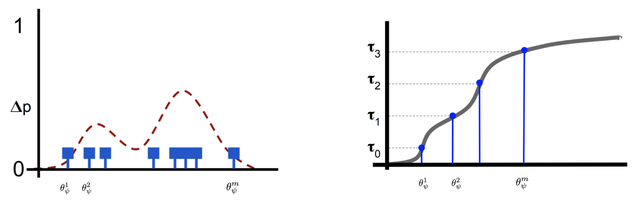
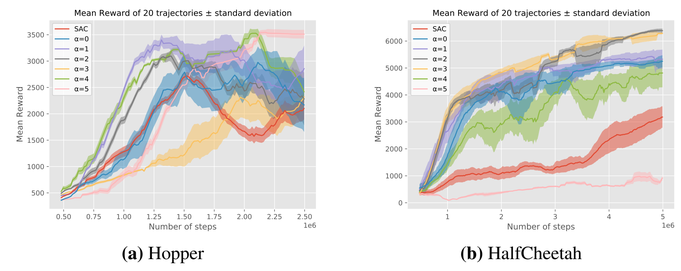


Figure 4: Quantile representation of the distribution of returns and cumulative distribution function.

5 Experiments

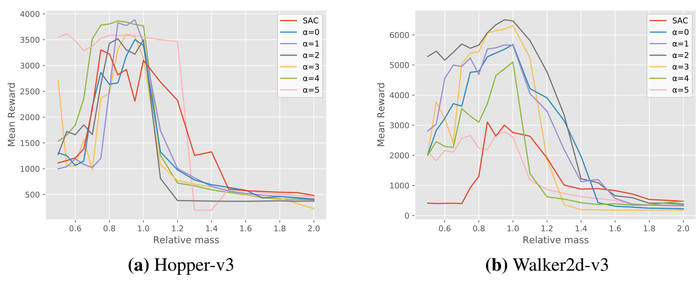
- Different experiments on continuous and discrete action space using $\xi : Z \to \mathbb{E}[Z] - \alpha \mathbb{V}[Z]^{\frac{1}{2}}$ instead of the mean.
- For continuous action space, we compare our algorithm with SAC in robust control [3].
- For discrete action space, we compare our approach with the classical PPO algorithm.

5.1 Continuous control



(a) Hopper **(b)** HalfCheetah

Figure 5: Mean performances over 20 trajectories ± variance.



(a) Hopper-v3 **(b)** Walker2d-v3

Figure 6: Mean over 20 trajectories varying relative mass of environments.

(a) Hopper-v3 **(b)** Walker2d-v3

Figure 7: Normalised mean over 20 trajectories varying relative mass of environments.

5.2 Discrete control

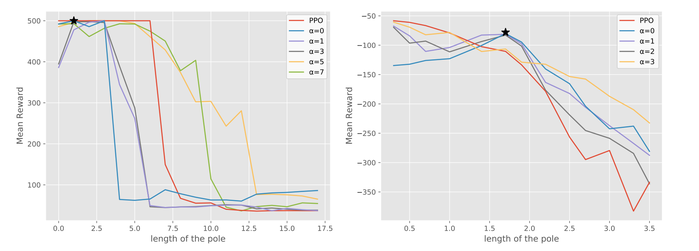


Figure 8: Mean over 20 trajectories varying length of the pole trained on the x-axis of the black star for Cartpole-v1 and Acrobot-v1 environments

Conclusions

- Simple risk averse RL as a robust RL problem.
- Improve robustness in discrete and continuous control.

Futur work :

- Proof of convergence (contraction and policy improvement theorem).
- Link with non distributional Robust Reinforcement Learning [2].

References

[1] Will Dabney, Mark Rowland, Marc G. Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pages 2892–2901, 10 2017.

[2] Esther Derman, Matthieu Geist, and Shie Mannor. Twice regularized mdps and the equivalence between robustness and regularization. *Advances in Neural Information Processing Systems*, 34, 2021.

[3] Benjamin Eysenbach and Sergey Levine. Maximum entropy rl (provably) solves some robust rl problems. *arXiv preprint arXiv:2103.06257*, 2021.

[4] Bruno Scherrer, Victor Gabillon, Mohammad Ghavamzadeh, and Matthieu Geist. Approximate modified policy iteration. *arXiv preprint arXiv:1205.3054*, 2012.