

Risk-sensitive Actor-free Policy via Convex Optimization

Ruoqi Zhang¹, Jens Sjölund¹

¹Uppsala University, Department of Information Technology, Box 337, SE-751 05, Uppsala, Sweden

Abstract

Traditional reinforcement learning methods optimize agents without considering safety, potentially resulting in unintended consequences. In this paper, we propose an *optimal* actor-free policy that optimizes a risk-sensitive criterion based on the conditional value at risk. The risk-sensitive objective function is modeled using an input-convex neural network ensuring convexity with respect to the actions and enabling the identification of globally optimal actions through simple gradient-following methods. Experimental results demonstrate the efficacy of our approach in maintaining effective risk control.

1. Introduction

Over the past decade, reinforcement learning (RL) has achieved notable advancements [1]. Nonetheless, traditional RL agents interact with their environment without accounting for safety, potentially leading to unintended and severe consequences in real-world applications. Safe RL addresses these concerns by ensuring that the learning process is both effective and safe. An intuitive way is to learn the policy subject to safety constraints, accounting for both parametric and inherent uncertainties within the model and its environment [2].

In this paper, we explore this approach by training a policy that optimizes a risk-sensitive criterion based on the conditional value at risk (CVaR). Traditional RL algorithms typically aim to maximize the expected value over all future cost-returns [3]. By emphasizing the tail of the future cost-return distribution, our learned policy explicitly penalizes infrequent occurrences of catastrophic events.

Additionally, we propose an actor-free architecture in which the action is implicitly defined as the solution to a convex optimization problem approximating the risk-sensitive criterion. This eliminates the need for incremental actor learning, which often necessitates hyperparameter tuning and tricks to stabilize the training process. With our actor-free approach, the policy aligns optimally with the approximated criterion. This deviates from prior research on CVaR-based safe RL [4, 5], which uses a neural network to approximate the actor. Key to our approach is to parameterize the risk-sensitive objective function using an input-convex neural network [6], ensuring convexity with respect to the actions (inputs). Consequently, simple gradient-following techniques can be used to find a globally optimal action.

2. Risk-sensitive Actor-free Policy

In this study, we focus on learning a risk-sensitive actor-free policy within a safety-constrained framework. The aim of the agent is to optimize future returns, maintaining compliance with safety cost constraints. The risk-sensitive objective function is structured using an input-convex neural network [6], guaranteeing convexity with respect to the actions (inputs). As a result, a globally optimal action can be identified.

2.1. Constrained Markov Decision Processes

We model the RL agent and its environment as a constrained Markov Decision Process (CMDP), represented by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, c, d, \gamma)$ where $\mathcal{S} \in \mathbb{R}^{d_S}$ is the state space, $\mathcal{A} \in \mathbb{R}^{d_A}$ is the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ is the probabilistic transition function, $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the immediate reward function, $c : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the immediate cost function, $d \in \mathbb{R}$ is the safety threshold and $\gamma \in (0.0, 1.0)$ is the discount factor.

The goal of the agent under the CMDP framework is to learn a policy that maximizes the expected return given an upper bound d on the (safety violation) cost,

$$\begin{aligned} & \underset{\pi}{\text{maximize}} && \mathbb{E}_{s_t \sim \rho_\pi, a_t \sim \pi(s_t)} \left[\sum_t \gamma^t r(s_t, a_t) \right] \\ & \text{subject to} && \mathbb{E}_{s_t \sim \rho_\pi, a_t \sim \pi(s_t)} \left[\sum_t \gamma^t c(s_t, a_t) \right] \leq d. \end{aligned} \quad (1)$$

where ρ_π is the stationary distribution over the state space \mathcal{S} under the policy π .

2.2. Safety Critic with Conditional Value at Risk

In CMDP, the safety violation costs are usually the (in)finite-horizon discounted future cost-return as shown in (1). However, only considering the expected value is insensitive to potentially hazardous events: policy gradient methods prefer a policy with lower cost, but also higher

The IJCAI-2023 AISafety and SafeRL Joint Workshop

ruoqi.zhang@it.uu.se (R. Zhang); jens.sjолund@it.uu.se

(J. Sjölund)

© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)



variance, over a policy with slightly higher cost but much lower variance. In other words, since higher variance amounts to higher risk, the policy is not risk-averse. To incorporate risk, we replace the expectation in the safety violation cost with the Conditional Value at Risk (CVaR) [7], a widely recognized risk measure that quantifies the amount of tail risk. More precisely, CVaR_α is defined as the expected reward of the worst α -percentile cases,

$$\Gamma_c^\pi(s, a, \alpha) \doteq \text{CVaR}_\alpha^\pi(C) = \mathbb{E}_{p_\pi} [C \mid C \geq F_C^{-1}(1 - \alpha)] \quad (2)$$

where $\alpha \in (0, 1]$ is used to define the risk level, C is a random variable and $F_C^{-1}(1 - \alpha)$ is the α -percentile. Calculating the CVaR measure directly for lengthy time horizons using, for example, sampling would be excessively costly [8]. Instead, we follow [5] and model the distribution of cost-return $C(s, a)$ as a Gaussian distribution $\mathcal{N}(Q_c^\pi(s, a), \sigma_c^2(s, a))$ where $Q_c^\pi(s, a) = \mathbb{E}_\pi[\sum_{i=t}^{\infty} \gamma^{i-t} c(s_i, a_i) \mid s_t = s, a_t = a]$ is the expected future cost-return and $\sigma_c^2(s, a)$ its variance. This Gaussian distribution leads to a closed-form CVaR measure of future cost-return [4, 5]

$$\Gamma_c^\pi(s, a, \alpha) = Q_c^\pi(s, a) + \frac{\phi(\alpha)}{\Phi(\alpha)} \sigma_c^\pi(s, a), \quad (3)$$

where ϕ is the standard normal distribution, and $\Phi(\cdot)$ is its CDF. Our risk-sensitive criterion can be written as,

$$\begin{aligned} & \underset{\pi}{\text{maximize}} && Q_r^\pi(s, a) \\ & \text{subject to} && Q_c^\pi(s, a) + \frac{\phi(\alpha)}{\Phi(\alpha)} \sigma_c^\pi(s, a) \leq d, \end{aligned} \quad (4)$$

where $Q_r^\pi(s, a) = \mathbb{E}_\pi[\sum_{i=t}^{\infty} \gamma^{i-t} r(s_i, a_i) \mid s_t = s, a_t = a]$ is the expected future return. To learn the mean and variance of cost-to-go, a distributional critic is learned with 2-Wasserstein distance as the loss function [5].

2.3. Optimal Actor-free Policy via Input Convex Neural Network

Policy gradient algorithms typically feature an actor-critic structure, utilizing two distinct neural networks known as the actor and the critic [9]. The critic estimates the reward-to-go or cost-to-go and the actor seeks to infer the action a to maximize the estimation from the critic. However, if the *optimal* action w.r.t the critic can be easily identified, the need for modeling the actor is eliminated. This can be achieved through parameterization of the reward with Partially Input Convex Neural Networks (PICNNs) [6]. We utilize two PICNNs, one to approximate $-Q_r(s, a)$ and another to estimate the cost-return distribution $Q_c(s, a)$ and $\sigma_c(s, a)$. In this way, the policy is *actor-free* since the optimal action a^* can be determined directly by minimizing

$$a^* = \arg \min_a -Q_r(s, a) + \kappa \max\{0, \Gamma_c(s_t, a_t) - d\} \quad (5)$$

where $\kappa \geq 0$ is a hyperparameter. This optimization problem is a nonsmooth exact penalty formulation of the constrained problem in (4). It is well-known that for sufficiently large κ the two problems have the same solution [10]. Moreover, since this problem is convex, finding a globally optimal action is tractable.

We first define the PICNN over state-action pairs $f(s, a; \theta)$ where f is convex in action a but not convex in state s . Figure 1 illustrates the simple convex network structure used in our paper. As shown in the figure, output z_3 can be calculated by forwarding the network,

$$\begin{aligned} u_0 &= s, & u_{i+1} &= \tilde{g}_i(\tilde{W}_i u_i + \tilde{b}_i), & i &= 0, 1 \\ z_1 &= g_0(W_0^{zs} s + W_0^{za} a + b_0), \\ z_{i+1} &= g_i(W_i^{zu} u_i + W_i^{za} a + W_i^{zz} z), & i &= 1, 2 \end{aligned} \quad (6)$$

where W are weight matrices, b are bias terms, g is the nonlinear activation function and z_3 is the output of the network which is made convex in the input a by restricting the weight matrices W_0^{za} and W_i^{zz} to be non-negative and the activation function g to be convex and non-decreasing, e.g. a rectified linear unit (ReLU).

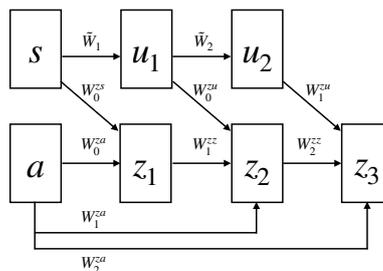


Figure 1: The partially input convex neural network (PICNN) structure used in the paper.

3. Experiment

The above method was evaluated in simulation on a continuous control task: cascade water tank level control. As depicted in Figure 2a, the task is to maintain a specific water level in the lower tank by changing the input signal a_t represents the voltage to the pump. The state of the system includes the heights of the two tanks, $s(t) = [s_1(t) \ s_2(t)]^T$ and the output is $y(t) = s_2(t)$. The reward function contains two parts, one related to the distance between the reference signal and the output, and the other to the cost incurred by a critical event. To introduce a risk element, we define a “critical” event as the level of the upper tank exceeding $l_{\text{crit}} = 10$ cm. Thus the reward and cost functions are defined as

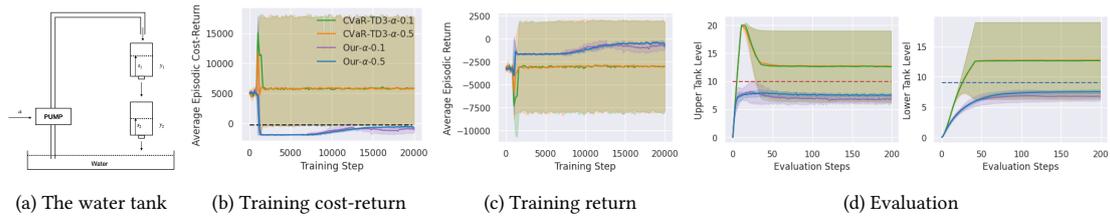


Figure 2: Performance comparison between our method with CVaR-TD3 with different risk values $\alpha = 0.1, 0.5$ during training with mean (solid lines) and ± 1 standard deviation (shaded area) of 5 runs. (The cost-return threshold d is in *dashed black*, the “critical” level is in *dashed red*, and the goal level is in *dashed blue*.)

$r(s, a) = -|s_2 - g|$, and $c(s, a) = s_1 - l_{\text{crit}}$, respectively. The long-term safety threshold d is set to -250 . The system was discretized using the Euler method in the simulation, with a sampling period of 2 seconds. Unlike a real-world tank, there are no upper bounds for s_1, s_2 in the simulation.

Our learning algorithm to update reward critic and safety critic is based on Twin Delayed Deep Deterministic policy gradient algorithm (TD3) [9] to avoid overestimating Q-values. As mentioned in Section 2.2, we use a reward critic and a distributional safety critic. Further, the policy is *actor-free*; the optimal action is found by a gradient descent algorithm, Adam [11].

We compare our method with its actor-critic version, CVaR-TD3, using a standard neural network instead of the PICNNs for actor, critic, and safety critic. The training results are illustrated in Figure 2 and Figure 2b with two metrics, average episodic returns, and average episodic cost-returns. It can be observed that CVaR-TD3 has a much higher variance, potentially attributed to the neural network-structured actor getting stuck in poor local minima, while our approach can identify the globally optimal action, aided by the PICNNs. Careful tuning of the hyperparameters could potentially mitigate this. Figure 2d shows the evaluation of the learned policy. To highlight the differences between the methods, the state s_1, s_2 is clipped with a maximum value 20. We observe that our method with $\alpha = 0.1$ exhibits a more conservative behavior, striving to maintain a safe distance from the critical level (*dashed red*). Consequently, our approach is also slightly farther from the goal level (*dashed blue*), which is close to the critical level.

4. Conclusions

We proposed a risk-sensitive actor-free policy with a CVaR criterion. The criterion is parameterized with input-convex neural networks ensuring convexity with respect to the actions. Thus, the globally optimal action can be found easily by simple gradient-descent methods. In the paper, future return and cost-return are approximated

by a Gaussian distribution in order to get a closed-form of CVaR. Future research could explore a more general distribution.

References

- [1] V. Mnih, et al., Human-level control through deep reinforcement learning, *Nature* 518 (2015) 529–533.
- [2] J. Garcia, F. Fernández, A comprehensive survey on safe reinforcement learning, *Journal of Machine Learning Research* 16 (2015) 1437–1480.
- [3] S. Ha, P. Xu, Z. Tan, S. Levine, J. Tan, Learning to walk in the real world with minimal human effort, in: *Conference on Robot Learning*, PMLR, 2021, pp. 1110–1120.
- [4] Q. Yang, T. D. Simão, S. H. Tindemans, M. T. Spaan, WCSAC: Worst-case soft actor critic for safety-constrained reinforcement learning, 2021.
- [5] Y. C. Tang, J. Zhang, R. Salakhutdinov, Worst cases policy gradients, in: *Conference on Robot Learning*, PMLR, 2020, pp. 1078–1093.
- [6] B. Amos, L. Xu, J. Z. Kolter, Input convex neural networks, in: *International Conference on Machine Learning*, PMLR, 2017, pp. 146–155.
- [7] P. Artzner, F. Delbaen, J.-M. Eber, D. Heath, Coherent measures of risk, *Mathematical finance* 9 (1999) 203–228.
- [8] A. Tamar, Y. Glassner, S. Mannor, Optimizing the CVaR via sampling, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015, pp. 2993–2999.
- [9] S. Fujimoto, H. Hoof, D. Meger, Addressing function approximation error in actor-critic methods, in: *International conference on machine learning*, PMLR, 2018, pp. 1587–1596.
- [10] S. Wright, J. Nocedal, et al., *Numerical optimization*, Springer Science 35 (1999) 7.
- [11] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: *3rd International Conference on Learning Representations, ICLR*, 2015.