POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Algorithms for Reward-based Coherent Risk Measures in Risk-Averse Reinforcement Learning

Laurea Magistrale in Computer Science and Engineering - Ingegneria Informatica

Author: Massimiliano Bonetti, 944205

Advisor: Prof. Marcello Restelli

Co-advisor: Dott. Lorenzo Bisi

Academic year: 2020-2021

1. Introduction

Reinforcement Learning (RL) methods have become very popular due to their ability of solving complex sequential decision making problems. In the classic risk-neutral stream of literature there are powerful solution methods like Trust Region Policy Optimization (TRPO) (Schulman et al., 2015) and Proximal Policy Optimization (PPO) (Schulman et al., 2017) that efficiently maximize the expected value of the cumulative discounted rewards (called expected return). Usually when dealing with realistic problems, like finance, robotics and healthcare, we want also to manage the *risk* in order to avoid bad events that can happen even if they are not very common. The risk can be split into three categories: *inherent risk*, which is due to the stochasticity of the environment; *model risk*, that refers to the imperfect knowledge of the environment which makes the consequences of its actions difficult to predict; *action risk*, which is due to the stochasticity of the actions done with intention by the agent, typically in order to do exploration. The action risk is under direct control of the agent. The inherent risk can be addressed by optimizing specific objective functions called risk measures, differently from the commonly used expected return. The model risk can be reduced using safe policy updates. Risk-averse Reinforcement Learning is not a new subject, several risk measures were introduced (Alexander et al., 2014) like: conditional value at risk (CVaR), variance-related measures, utility function, entropic risk measure. More interesting are the coherent risk measures (Alexander et al., 2014; Tamar et al., 2015), that are characterized by convexity, monotonicity, translation equivariance and positive homogeneity. These properties allow for example to obtain solutions that are more rational like avoiding policies that always give the lowest possible reward. Furthermore, in Bisi et al. (2020) was introduced a new risk measure based on the reward instead of the return: the Mean-Volatility, which smooths the trajectories avoiding shocks. In this work we want to capture the advantages of the coherence properties and of the reward-based measures by introducing two new risk measures that are both coherent and reward-based: the Mean-RMAD and the RCVaR, where the Mean is the normalized expected return, the RMAD stands for Reward-based Mean Absolute Deviation and RCVaR is the Reward-based Conditional Value at Risk. Furthermore, we provide safe updates, thanks to the Performance Difference Lemma that allows to develop a TRPO-like algorithm for both measures with guaranteed monotonic improvement. For the RCVaR we can use also any risk-neutral Reinforcement Learning algorithm.

We recall some concepts about the risk measures in Section 2. In Section 3 we introduce and motivate the use of the Mean-RMAD and of the RCVaR. In Section 4 we show RMAD-TRPO and RMAD-PPO, while Section 5 is dedicated to the algorithms for the RCVaR. Finally, in Section 6, we conduct an empirical analysis of the new algorithms on a financial environment, on the challenging robotic environment Hopper from PyBullet and on an easier environment called *Point Reacher*.

2. Preliminaries

Mathematical Background. Given a measurable space $(\mathcal{X}, \sigma_{\mathcal{X}})$, where $\mathcal{X}$ is a set and $\sigma_{\mathcal{X}}$ a σ-algebra, we denote with $\Delta_{\mathcal{X}}$ the set of probability measures and with $\mathscr{B}(\mathcal{X})$ the set of bounded measurable functions. Given any probability $\mathcal{P} \in \Delta_{\mathcal{X}}$, $\mathbb{P} = (\mathcal{X}, \sigma_{\mathcal{X}}, \mathcal{P})$ is a probability space. On this space we define uncertain outcomes $Z = Z(x)$, which are random functions over the outcomes $x \in \mathcal{X}$. The space $\mathcal{Z}_{\mathcal{X}}$ of allowable random functions Z we deal with is $\mathcal{Z} \coloneqq \mathcal{L}_p(\mathcal{X}, \sigma_{\mathcal{X}}, \mathcal{P})$, where $p \in [1, \infty)$, so the random variable Z has a finite p-th order moment. We denote with $\succeq$ a *pointwise partial order* over $\mathcal{Z}$: given $Z, Z' \in \mathcal{Z}$, $Z \succeq Z'$ means that $Z(x) \geqslant Z'(x)$ for $\mathcal{P}$-almost all $x \in \mathcal{X}$.

Markov Decision Processes. We consider discrete-time, discounted Markov Decision Process (MDP) with infinite time-horizon. An MDP is a tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma, \mu)$, where:

- $\mathcal{S}$ is the (continuous) state space;
- $\mathcal{A}$ is the (continuous) action space;
- $P(\cdot|s,a) \in \Delta_{\mathcal{S}} \quad \forall s \in \mathcal{S}, a \in \mathcal{A}$ is the Markovian transition kernel, indicating the probability of reaching a specific state when performing action a in state s;
- $R : \mathcal{S} \times \mathcal{A} \to [-R_{max}, R_{max}]$ is the bounded reward function;

- $\gamma \in (0,1)$ is the discount factor;
- $\mu(\cdot) \in \Delta_{\mathcal{S}}$ is the starting-state distribution.

The agent's behaviour is determined by a Markovian, stationary policy, defined as the mapping $\pi : \mathcal{S} \to \Delta_{\mathcal{A}}$, where $\pi(a|s)$ is the probability of performing action a while being on state s. We will sometimes restrict our attention to parametric policies $\pi_{\boldsymbol{\theta}} \in \Pi_\Theta$ identified by a vector of parameters $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^m, m \geqslant 1$.

Following policy π, the interaction between the agent and the environment determines a trajectory $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, ...)$, where $s_0 \sim \mu(\cdot)$, $a_t \sim \pi(\cdot|s)$, $r_t = R(s_t, a_t)$ and $s_{t+1} \sim P(\cdot|s_t, a_t)$ for all $t \geqslant 0$. We call $\mathcal{T}$ the set of all possible trajectories. A trajectory is a *random variable* whose probability density, given some policy π, is:

$$p_\pi(\tau) = \mu(s_0) \prod_{t=0}^{\infty} \pi(a_t|s_t) P(s_{t+1}|s_t, a_t).$$

Given a trajectory τ followed by the agent, the discounted cumulative reward is called *return* and it is defined as $G(\tau) \coloneqq \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)$. The expectations of the return $\mathbb{E}_{\tau|\pi}[G(\tau)]$ conditioned to $s_0 = s$ and $s_0 = s, a_0 = a$ are the *value function* $V_\pi(s)$ and the *action value function* $Q_\pi(s,a)$, respectively. The gain of choosing action a in state s is given by the *advantage function*: $A_\pi(s,a) \coloneqq Q_\pi(s,a) - V_\pi(s)$. When considering parametric policies, it is convenient to evaluate the agent performance w.r.t. a scalar criterion called *expected return* $J_\pi \coloneqq \mathbb{E}_{s_0 \sim \mu(\cdot)}[V_\pi(s_0)] = \mathbb{E}_{\tau \sim p_\theta(\cdot)}[G(\tau)]$.

The (discounted) state occupancy measure induced by policy π is defined as:

$$d_{\mu,\pi}(s) \coloneqq (1-\gamma) \int_{\mathcal{S}} \mu(s_0) \sum_{t=0}^{\infty} p_\pi(s_0 \xrightarrow{t} s) \mathrm{d}s_0,$$

where $p_\pi(s_0 \xrightarrow{t} s)$ is the probability of reaching state s after t steps starting from state s_0 and following policy π.

The expectation of the reward w.r.t. the state occupancy measure is called *normalized expected return* (sometimes we will use *n. expected return* for short): $\bar{J}_\pi \coloneqq \mathbb{E}_{\substack{s \sim d_{\mu,\pi}(\cdot) \\ a \sim \pi(\cdot|s)}}[R(s,a)] = (1-\gamma) J_\pi$.

Risk-Measures.

Definition 2.1 (Risk-Measure, (Alexander et al., 2014)). *Given a probability space $(\mathcal{X}, \sigma_{\mathcal{X}}, \mathcal{P})$, and some uncertain outcome $Z \in \mathcal{Z}_{\mathcal{X}}$, we call* risk measure *a function which maps Z into the extended real line $\bar{\mathbb{R}} = \mathbb{R} \cup \{+\infty\} \cup \{-\infty\}$.*

In order to guarantee that optimizing a risk-measure induces a *rational* behavior, such measure needs to respect some axioms.

Definition 2.2 (Coherent Risk-Measure). *A risk-measure η, defined w.r.t. the uncertain outcome Z, is* coherent *if it satisfies the following properties for all $Z, Z' \in \mathcal{Z}$:*

- Concavity[1]*:* $\eta(tZ + (1-t)Z') \geqslant t\eta(Z) + (1-t)\eta(Z') \quad \forall t \in [0,1]$.
- Monotonicity*: If $Z \succeq Z'$, then $\eta(Z) \geqslant \eta(Z')$.*
- Translation Equivariance*:* $\forall a \in \mathbb{R} : \eta(Z+a) = \eta(Z) + a$.
- Positive Homogeneity*:* $\forall t > 0 : \eta(tZ) = t\eta(Z)$.

Risk-Averse Reinforcement Learning. We consider a risk-averse optimization reinforcement learning context, in which the agent does not seek to maximize the risk-neutral objective J_π, but a risk-averse variant of it, typically a risk-measure. We introduce the following distinction among the risk-measures, which classifies them according to the considered probability spaces.

Definition 2.3 (Return-based and Reward-based Measures). *A risk-measure defined w.r.t. a probability space $\mathbb{P}$ and an uncertain outcome Z is called:*

- return-based*, if $\mathbb{P} = (\mathcal{T}, \sigma_{\mathcal{T}}, p_\pi)$, and $Z = G(\tau)$;*
- reward-based*, if $\mathbb{P} = (\mathcal{S} \times \mathcal{A}, \sigma_{\mathcal{S}\times\mathcal{A}}, d_{\mu,\pi})$, and $Z = R(s,a)$.*

As discussed in Bisi (2022), the return-based risk measures can capture only the risk on the return, thus they are insensitive to short-term risk; while the reward-based risk measures captures short-term risk because they consider the per-step reward, smoothing the trajectories that avoid shocks. The solely reward-based risk-averse objective analysed in literature so far is the Mean-Volatility (Bisi et al., 2020), where the Volatility is:

$$\nu_\pi^2 \coloneqq \mathop{\mathbb{E}}_{\substack{s \sim d_{\mu,\pi}(\cdot) \\ a \sim \pi(\cdot|s)}} \left[\left(R(s,a) - \bar{J}_\pi \right)^2 \right].$$

3. Coherent Reward-based Risk-Averse Objectives

3.1. Mean-RMAD and RCVaR

Definition 3.1 (Mean-RMAD). *Given the probability space $\mathbb{P} = (\mathcal{S} \times \mathcal{A}, \sigma_{\mathcal{S}\times\mathcal{A}}, d_{\mu,\pi})$, we define the* reward-based mean absolute deviation (RMAD) *as:*

$$\omega_\pi \coloneqq \mathop{\mathbb{E}}_{\substack{s \sim d_{\mu,\pi}(\cdot) \\ a \sim \pi(\cdot|s)}} \left[\left| R(s,a) - \bar{J}_\pi \right| \right].$$

By setting a risk-aversion factor λ, we can define also a trade-off measure called reward-based Mean-MAD (Mean-RMAD)*: $\eta_\pi^\lambda \coloneqq \bar{J}_\pi - \lambda \omega_\pi$.*

It includes the classic mean and it penalizes the deviations from the mean, that can cause high variability of the results. Differently from the variance, the MAD doesn't square the deviations, but they affect only with the distance from the mean, while the variance weights more deviations bigger than one and less the deviations smaller than one.

Definition 3.2 (RCVaR). *Given $\mathbb{P} = (\mathcal{S} \times \mathcal{A}, \sigma_{\mathcal{S}\times\mathcal{A}}, d_{\mu,\pi})$ and $\alpha \in (0,1)$, we define the* reward-based conditional value-at-risk (RCVaR) *as:*

$$\eta_\pi^\alpha \coloneqq \max_\rho \left\{ \rho - \frac{1}{\alpha} \mathop{\mathbb{E}}_{\substack{s \sim d_{\mu,\pi}(\cdot) \\ a \sim \pi(\cdot|s)}} \left[(R(s,a) - \rho)_- \right] \right\},$$

and the value ρ_π^α for which the above program is optimal is what we call the reward-based value-at-risk (RVaR)[2]*.*

The RCVaR (like the CVaR) captures the mean of the worst outcomes, in this way its optimization reduces the bad events. The RCVaR is coherent $\forall \alpha \in (0,1)$, while the Mean-MAD is coherent whenever $0 \leqslant \lambda \leqslant 0.5$ (Alexander et al., 2014). The following proposition relates the introduced reward-based risk measures with their corresponding return-based versions. Given the probability space $\mathbb{P} = (\mathcal{T}, \sigma_{\mathcal{T}}, p_\pi)$, we just need to recall the mean absolute deviation:

$$\omega_\pi^G \coloneqq \mathop{\mathbb{E}}_{\tau|\pi} \left[|R(s,a) - J_\pi| \right]$$

and the CVaR:

$$\eta_\pi^{\alpha,G} \coloneqq \max_\rho \left\{ \rho - \frac{1}{\alpha} \mathop{\mathbb{E}}_{\tau|\pi} \left[(R(s,a) - \rho)_- \right] \right\}.$$

[1] In the minimization formulation we have to substitute this property with *convexity*.

[2] We drop the dependence from π whenever it is clear from the context.

Proposition 3.1. *The following relationships hold between reward-based and return-based risk-measures:*

$$\omega_\pi^G \leq \frac{\omega_\pi}{(1-\gamma)}, \quad \eta_\pi^{\alpha,G} \geq \frac{\eta_\pi^\alpha}{(1-\gamma)}.$$

This result tells us that optimizing reward-based risk measures amounts to bound the corresponding return-based risk measure, similarly to what happens for the reward-volatility (Bisi et al., 2020). It is also possible to establish relationships among reward-based measures.

Proposition 3.2. *Given an MDP $\mathcal{M}$ and a policy π, $\forall \alpha \in (0,1)$, the following relationships hold among reward-based risk-measures:*

$$\eta_\pi^\alpha \geq \eta_\pi^\lambda \quad \textit{if} \quad \lambda = \frac{1}{2\alpha},$$
$$(\omega_\pi)^2 \leq \nu_\pi^2.$$

Interestingly, the previous proposition lower bounds the RCVaR value of some policy w.r.t. the Mean-RMAD one. Therefore, optimizing the latter quantity can also see as a proxy to optimize the former one. In Figure 1, we see that the optimal policies of the Mean-RMAD are different from the optimal policies of the RCVaR, obtained with brute force on the environment *Point Reacher* (Bisi, 2022). The same happens also if we compare the Mean-MAD with the Mean-Volatility and the CVaR with the Mean-Volatility. It means that they capture different preferences w.r.t. risk, thus, the best risk measure to use in practice may depend on the problem at hand.

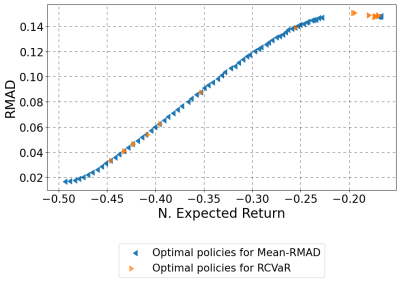


Figure 1: Comparison between the optimal policies of Mean-RMAD and of RCVaR, obtained with brute force on the environment *Point Reacher* (Bisi, 2022).

Our choice of analysing these two risk-measures is motivated by the particular combination of properties they enjoy. We compare in Table 1 the measures in exam with state-of-the-art measures. The dimensions under which we analyse them involve both coherence features and reinforcement learning properties. We show that the chosen risk-measures, beyond being coherent, enjoy expectation Bellman equations and a formulation of the Performance Difference Lemma. These features are fundamental for the development of *policy gradient* approaches, *safe improvement bounds* and effective *trust-region* approaches. Looking at the table, it is possible to notice that RCVaR and Mean-RMAD share indeed these properties with Mean-Volatility, which is not coherent though, due to the lack of the monotonicity property.

Risk Measure	Coherency	Bellman equations	PDL
Mean-RMAD and RCVaR	✓	✓	✓
Mean-Volatility	X	✓	✓
Mean-Variance	X	✓	NK
CVaR	✓	✓	NK
Utility model	X	NK	NK
Entropic risk measure	X	✓	NK

Table 1: Properties of various risk measures. In gold there are the new risk measures introduced in this document. "NK" means Not Known.

3.2. The Importance of Coherence: a Motivating Example

Risk-measures have originally been developed by the financial literature as a way of computing the necessary amount of cash that need to be reserved to shield against some potential risk. In this context, the coherence axioms have been selected in order to guarantee a rational behavior: *Concavity*, it ensures that diversification is always beneficial to risk; *Monotonicity*, it makes choices resulting in lower reward for each outcome riskier. It avoids risk-averse optimization to converge to degenerate solutions; *Translation Equivariance*, it allows to exclude deterministic components from risk computation. In a reinforcement learning perspective, it also allow reward translation by a constant quantity, which may be beneficial in some tasks in order to enhance exploration; *Positive Homogeneity*, it encodes the intuition that multiplying the exposition directly maps to risk. From a reinforcement learning viewpoint, this property allows reward scaling, which may be useful in practice in case of extreme reward ranges (very low, or very high) to avoid precision or overflow errors. By looking at Table 1, it is possible to notice that important risk-measures as Mean-Variance, Mean-Volatility and Entropic Risk-Measure (ERM) are not coherent. Therefore, optimizing these risk-averse objective may result in irrational behaviors. Violating the monotonicity is of particular concern, since it permits the agent to possibly converge to solutions which should be excluded instead. With the following example we will try to provide the reader some further intuition on this point.

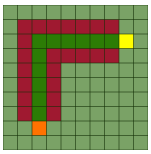
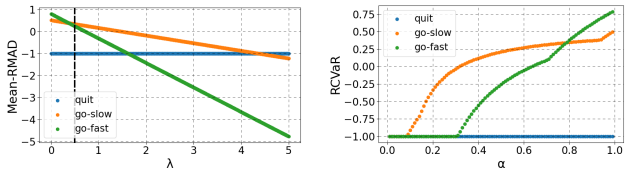


Figure 2: Graphical representation of the Grid-World Garden environment. The orange square is the starting-state, while the yellow square is the goal-state in which the agent receives the highest reward.



(a) Mean-RMAD (b) RCVaR

Figure 3: Evaluation on the environment Garden of three policies: the go-fast policy is the one that reaches the goal state with high speed, the go-slow policy goes to the goal state with low speed, while the idle policy is the one that goes on the grass. The vertical dashed line in Figure 3a indicates that over that value the risk measure is no more coherent.

The Garden Example. Consider a real-world scenario in which a gardener-agent has to learn how to cut a hedge and it must avoid collisions with flowers or people walking on the garden grass. A pictorial representation of a simplified $2D$ model with stochastic transitions is given in Figure 2. The agent can control his direction and speed, with higher speeds increasing the chance of losing control over its direction. Positive reinforce is provided for approaching and reaching some goal state. Importantly, the lowest possible reward is achieved if the agent enters the grass, where people may be hit. We considered three policies:

- *go-fast*: it follows the path to the goal-state with high speed (two steps per action);
- *go-slow*: it follows the path to the goal-state with low speed;
- *quit*: it quits the task prematurely, by immediately touching the grass.

We notice that the *quit* policy has the worse risk-neutral performance, giving always the lowest possible reward. On the other hand, by instantly quitting the task, this behaviour allows to obtain a low variability for the reward, hence, it may be preferred by an extremely risk-averse agent. We recall that, entering the grass, the gardener agent might risk to hurt the people in the area surrounding the hedge, a behaviour that we explicitly tried to discourage by providing a high penalty. Figures 3a and 3b show the performance of each policies according to, respectively, Mean-RMAD and RCVaR. Thanks to the monotonicity property, the Mean-RMAD for $0 \leq \lambda \leq 0.5$ and the RCVaR consider the *quit* policy as not the best. However, it can be noticed that it is selected for higher level of λ, the reason is that monotonicity is no longer guaranteed. Therefore, such risk-aversion levels may induce convergence of some learning algorithm towards dangerous policies as a byproduct of an excessive aversion to risk. While this phenomenon can be avoided a priori for Mean-RMAD by limiting the range of λ, the same cannot be done with Mean-Volatility, hence, unwanted behaviors can only be spotted a posteriori.

Another key tool for making online algorithms reliable is to provide *safe* guarantees to ensure a stable performance improvement, as it is shown in the next sections.

4. Mean-RMAD Optimization

4.1. Value functions

We introduce the value function $W_\pi(s)$ and the action value function $X_\pi(s,a)$ of the RMAD, obtained from:

$$\mathop{\mathbb{E}}_{\substack{a_t\sim\pi(\cdot|s_t)\\ s_{t+1}\sim P(\cdot|s_t,a_t)}}\left[\sum_{t=0}^{\infty}\gamma^t\left|R(s_t,a_t)-\bar{J}_\pi\right|\right]$$

by conditioning on $s_0 = s$ and $s_0 = s, a_0 = a$, respectively. The value functions of the Mean-RMAD can then be obtained as a linear combination of the classic risk-neutral value functions and the RMAD value functions:

$$V_\pi^\lambda(s) \coloneqq V_\pi(s) - \lambda W_\pi(s),$$
$$Q_\pi^\lambda(s,a) \coloneqq Q_\pi(s,a) - \lambda X_\pi(s,a).$$

The definition of the advantage functions of the RMAD and of the Mean-RMAD automatically follow, respectively:

$$A_\pi^\omega(s,a) \coloneqq -(X_\pi(s,a) - W_\pi(s)),$$
$$A_\pi^\lambda(s,a) \coloneqq Q_\pi^\lambda(s,a) - V_\pi^\lambda(s) = A_\pi(s,a) + \lambda A_\pi^\omega(s,a),$$

where we included the minus sign because we prefer lower values of the RMAD.

4.2. Safe Improvement Guarantees and Trust-Region Algorithms

Monotonic Performance Improvement. We extend the Performance Difference Lemma to the Mean-RMAD in order to develop a trust region method (Schulman et al., 2015).

Lemma 4.1 (Mean-RMAD Performance Difference Lemma). *The difference of the performance in terms of Mean-RMAD between two policies π and $\tilde{\pi}$ is lower bounded by:*

$$\frac{\eta_{\tilde{\pi}}^\lambda - \eta_\pi^\lambda}{(1-\gamma)} \geqslant \mathop{\mathbb{E}}_{\tau|\tilde{\pi}}\left[\sum_t \gamma^t A_\pi^\lambda(s_t,a_t)\right] - \lambda\left|\mathop{\mathbb{E}}_{\tau|\tilde{\pi}}\left[\sum_t \gamma^t A_\pi(s_t,a_t)\right]\right|.$$

From this lemma we obtain the safe improvement bound for parametric policies in Theorem 4.1.

Theorem 4.1 (Mean-RMAD Safe Improvement Bound). *Consider the following approximation of $\eta_{\pi_{\boldsymbol{\theta}}}^\lambda$, replacing the state-occupancy density of the old policy $d_{\mu,\pi_{\boldsymbol{\theta}_k}}$:*

$$L_k^\lambda(\pi_{\boldsymbol{\theta}}) \coloneqq \eta_{\pi_{\boldsymbol{\theta}_k}}^\lambda + \int_{\mathcal{S}} d_{\mu,\pi_{\boldsymbol{\theta}_k}}(s)\int_{\mathcal{A}}\pi_{\boldsymbol{\theta}}(a|s)A_{\boldsymbol{\theta}_k}^\lambda(s,a)\,da\,ds.$$

Then, the performance of $\pi_{\boldsymbol{\theta}}$ can be bounded as follows:

$$\eta_{\pi_{\boldsymbol{\theta}}}^\lambda \geqslant L_k^\lambda(\pi_{\boldsymbol{\theta}}) - \frac{4\gamma\epsilon_\lambda}{1-\gamma}\alpha_{KL}^2 - \lambda(1-\gamma)M,$$

where:

$$\alpha_{KL}^2 = \max_s D_{KL}(\pi_{\boldsymbol{\theta}_k}(\cdot|s),\pi_{\boldsymbol{\theta}}(\cdot|s)),$$
$$\epsilon_\lambda = \max_{s,a}\left|A_{\boldsymbol{\theta}_k}^\lambda(s,a)\right|,\quad \epsilon = \max_{s,a}\left|A_{\boldsymbol{\theta}_k}(s,a)\right|,$$
$$M \coloneqq |A_{\boldsymbol{\theta}_k}^{\boldsymbol{\theta}}| + \frac{4\epsilon\gamma}{(1-\gamma)^2}\alpha_{KL}^2,$$
$$A_{\boldsymbol{\theta}_k}^{\boldsymbol{\theta}} \coloneqq \mathop{\mathbb{E}}_{\tau|\pi_{\boldsymbol{\theta}_k}}\left[\sum_{t=0}^{\infty}\gamma^t \mathop{\mathbb{E}}_{a\sim\pi_{\boldsymbol{\theta}}(\cdot,s_t)}[A_{\boldsymbol{\theta}_k}(s_t,a)]\right]$$

and D_{KL} is the Kullback-Leibler divergence.

By optimizing the Safe Improvement Bound we get RMAD-TRPO, described in Algorithm 1.

Algorithm 1 RMAD-TRPO

1: **Input:** initial policy parameter $\boldsymbol{\theta}_0$, batch size N, number of iterations K, discount factor γ.
2: **for** $k = 0, ..., K-1$ **do**
3: Collect N trajectories with $\boldsymbol{\theta}_k$.
4: Compute advantage values $A_{\boldsymbol{\theta}_k}^\lambda(s,a)$ and $A_{\boldsymbol{\theta}_k}^{\boldsymbol{\theta}}(s,a)$.
5: Solve the constrained optimization problem:

$$\boldsymbol{\theta}_{k+1} = \arg\max_{\boldsymbol{\theta}\in\Theta}\left\{L_k^\lambda(\pi_{\boldsymbol{\theta}}) - \frac{4\gamma\epsilon_\lambda}{1-\gamma}\alpha_{KL}^2 - \lambda(1-\gamma)M\right\},$$

where $L_k^\lambda(\pi_{\boldsymbol{\theta}})$, M, $A_{\boldsymbol{\theta}_k}^{\boldsymbol{\theta}}$, ϵ_λ, ϵ and α_{KL}^2 are defined in Theorem 4.1.
6: **end for**

The TRPO version for the Mean-RMAD has the same guarantee of monotonic improvement (Corollary 4.1) of the original risk-neutral method.

Corollary 4.1 (Monotonic Improvement of RMAD-TRPO). *By optimizing the Mean-RMAD Safe Improvement Bound of Theorem 4.1 at each iteration k, we obtain a monotonic improvement of the Mean-RMAD:*

$$\eta_{\pi_{k+1}}^\lambda \geqslant \eta_{\pi_k}^\lambda \quad \forall k \geqslant 0,$$

where $\eta_{\pi_k}^\lambda$ is the Mean-RMAD of policy π_k at iteration k.

For the practical version of RMAD-TRPO we followed Schulman et al. (2015), using a constraint on the Kullback-Leibler divergence instead of a penalty.

RMAD-PPO. We can create a version of PPO (Schulman et al., 2017) for the Mean-RMAD objective using the fact that one subgradient with respect to θ of the Mean-RMAD objective is equal to the gradient of:

$$(1-\gamma)\mathop{\mathbb{E}}_{\tau\sim p_{\pi_{old}}}\left[\sum_{t=0}^{\infty}\gamma^t\frac{\pi_{\boldsymbol{\theta}}(a_t|s_t)}{\pi_{\boldsymbol{\theta}_{old}}(a_t|s_t)}A_{\pi_{old}}^{\lambda,\psi}(s_t,a_t)\right],$$

where we used the advantage function of the transformed reward $\tilde{R}(s,a) \coloneqq R(s,a) - \lambda\left|R(s,a) - \bar{J}_\pi\right| + \lambda\psi_\pi R(s,a)$:

$$A_\pi^{\lambda,\psi}(s_t,a_t) \coloneqq A_\pi(s_t,a_t) + \lambda A_\pi^\omega(s_t,a_t) + \lambda\psi_\pi A_\pi(s_t,a_t).$$

RMAD-PPO remains the same as PPO, we just need to substitute the risk-neutral advantage function of PPO with $A_\pi^{\lambda,\psi}$.

5. RCVaR Optimization

5.1. Decomposition and Optimization via risk-neutral RL methods

In order to optimize the RCVaR we exchange the maximization with respect to $\boldsymbol{\theta}$ with the maximization with respect to ρ:

$$\max_{\boldsymbol{\theta}\in\Theta} \eta^{\alpha}_{\pi_{\boldsymbol{\theta}}} = \max_{\rho} \max_{\boldsymbol{\theta}\in\Theta} (1-\gamma) \mathop{\mathbb{E}}_{\substack{s_0\sim\mu(\cdot),\\ a_t\sim\pi_{\boldsymbol{\theta}}(\cdot|s_t)\\ s_{t+1}\sim P(\cdot|s_t,a_t)}} \left[\sum_{t=0}^{\infty}\gamma^t\left(\rho - \frac{1}{\alpha}\left(R(s,a)-\rho\right)_-\right)\right], \quad (1)$$

which can be solved in a block-coordinate fashion. For a fixed ρ, the inner problem is an MDP with the transformed reward $\tilde{R}(s,a) = \rho - \frac{1}{\alpha}\left(R(s,a)-\rho\right)_-$, that allows to apply any risk-neutral RL method. While for a fixed policy $\pi_{\boldsymbol{\theta}}$, the solution of the outer problem is the RVaR. Our approach is described in Algorithm 2, it alternates the calculation of the RVaR with the optimization of an MDP. Unfortunately, we have not found a similar decomposition for the Mean-RMAD, but future work may investigate more its properties. Our algorithm is similar to Risk-Averse policy Optimization by State Augmentation (ROSA) of (Bisi, 2022), where the exchange between the maximization with respect to the policy and the maximization with another variable allows to apply any risk-neutral RL algorithm to a sequence of MDP. Its inner optimization problem is not an MDP, so ROSA requires also the augmentation of the state at each iteration, that may cause an increase of the sample complexity, while our method doesn't require it.

Algorithm 2 RCVaR Block cyclic coordinate ascent (RCVaR-BCCA)

1: **Input:** initial policy parameter $\boldsymbol{\theta}_0$, batch size N, number of iterations K, discount factor γ, risk-neutral RL algorithm A (e.g. PPO, TRPO, etc.)).
2: **for** $k = 0, ..., K-1$ **do**
3: Collect a batch $\{\tau_i\}_{i=1}^{N}$ of N trajectories with $\pi_{\boldsymbol{\theta}_k}$.
4: Compute the RVaR $\rho^{\alpha}_{\pi_{\boldsymbol{\theta}_k}}$.
5: Feed $\{\tau_i\}_{i=1}^{N}$ into A and maximize the inner problem 1 with respect to $\boldsymbol{\theta}\in\Theta$, by fixing $\rho = \rho^{\alpha}_{\pi_{\boldsymbol{\theta}_k}}$, to obtain $\pi_{\boldsymbol{\theta}_{k+1}}$.
6: **end for**

RCVaR-BCCA is characterized by the monotonic improvement guarantee of the RCVaR.

Theorem 5.1 (Monotonic Policy Improvement for block cyclic coordinate ascent). *Following Algorithm 2, the RCVaR grows monotonically:*

$$\eta^{\alpha}_{\pi_{\boldsymbol{\theta}_{k+1}}} \geqslant \eta^{\alpha}_{\pi_{\boldsymbol{\theta}_k}} \quad \forall k \geqslant 0,$$

where $\eta^{\alpha}_{\pi_{\boldsymbol{\theta}_k}}$ is the RCVaR of policy $\pi_{\boldsymbol{\theta}_k}$ at iteration k.

In the experiments we used RCVaR-BCCA with TRPO and PPO, which we call RCVaR-TRPO, respectively. The reason is that RCVaR-TRPO allows safe updates that reduce the model risk and because TRPO and PPO are able to tackle complex and large-scale control problems.
RCVaR-TRPO can be obtained also by developing the performance difference lemma for the RCVaR, from which it is possible to get a safe improvement bound that can be maximized with guaranteed monotonic improvement. The performance difference lemma and the safe improvement are equal to those of the risk-neutral TRPO, instead of the risk-neutral advantage function they need only the following one, defined over the transformed reward $\tilde{R}(s,a) := -\frac{1}{\alpha}\left(R(s,a)-\rho\right)_-$,:

$$A^{\rho}_{\pi}(s,a) := Q^{\rho}_{\pi}(s,a) - V^{\rho}_{\pi}(s),$$

where the value function $V^{\rho}_{\pi}(s)$ and the action value function $Q^{\rho}_{\pi}(s,a)$ come from:

$$\mathop{\mathbb{E}}_{\substack{a_t\sim\pi(\cdot|s_t)\\ s_{t+1}\sim P(\cdot|s_t,a_t)}} \left[\sum_{t=0}^{\infty}\gamma^t\left(-\frac{1}{\alpha}\left(R(s,a)-\rho\right)_-\right)|s_0=s\right]$$

by conditioning on $s_0 = s$ and $s_0 = s, a_0 = a$, respectively.

6. Experiments

We performed an empirical analysis of the algorithm developed in the previous sections: RMAD-TRPO (Algorithm 1), RMAD-PPO (Section 4.2), RCVaR-TRPO (Algorithm 2 with TRPO) and RCVaR-PPO (Algorithm 2 with PPO), and compared them with TRVO (Bisi et al., 2020) in terms of learning speed and quality of the retrieved approximated Pareto frontier. The objective is to show the risk-sensitivity, the trade-off between the normalized expected return and the risk, the speed of convergence and the ability to optimize the considered risk measure. The results are the average of 5 independent runs and in each environment we considered 5 risk-aversion levels for each risk measure.

6.1. Noisy Point Reacher

We consider a modified version of *Point Reacher* (Bisi, 2022) with more noise, in which the agent controls a point mass that moves along the real line in order to bring it to a target location in the minimum number of steps. The goal is to move the point as near as possible to the origin, but the higher the speed the higher the risk.

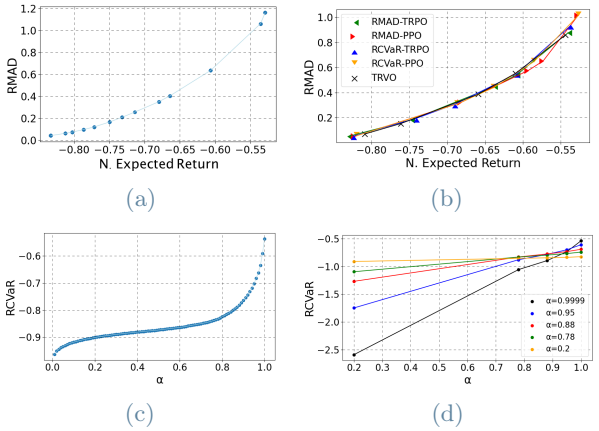


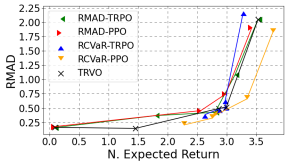
Figure 4: Results on Noisy Point Reacher. In Figures 4a and 4c there are the optimal values for the Mean-RMAD and for the RCVaR, respectively, obtained with brute force. Figure 4b shows the trade-off Mean RMAD obtained by the algorithms indicated in the legend. Figures 4d displays the RCVaR for different values of α of the policies trained with RCVaR-TRPO, which tried to optimize the RCVaR with α indicated in the legend. The lines that connect the points are showed only for readability.

In Figure 4a and 4c there are the optimal policies obtained with brute force. The obtained frontier in Figure 4b approximate the frontiers composed by the optimal policies in Figure 4a. The results of Figures 4d come from RCVaR-TRPO (similar results were obtained with RCVaR-PPO) and we can see that for $\alpha = 0.2$ the highest RCVaR is achieved by the policy that tried to maximize the RCVaR with $\alpha = 0.2$; while for $\alpha = 0.78$ the best policy is the one that tried to maximize the RCVaR with $\alpha = 0.78$ and so on with the other values of α. Furthermore, the highest values of the RCVaR for each α of Figure 4d are similar to the optimal values obtained with brute force in Figure 4c. These results

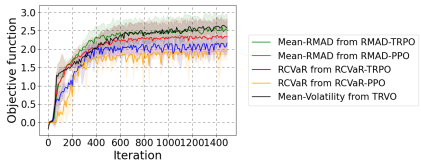
indicate that the algorithms have found policies that have a performance very similar to the optimal one.

6.2. Hopper

We considered one challenging environment in the robotic setting: Hopper from PyBullet. The state of the robot is made up of its position and its speed, while the actions consists of torques applicable to various joints. The state space and the action space are continuous and high-dimensional. The reward is equal to a linear combination of: a bonus for being alive, a bonus for its distance from the initial position, a cost for large actions in absolute value and a cost if the joints of the robot are at their limit. The dynamics is deterministic so to obtain a sensible environment for risk-averse optimization we added noise with zero mean to the actions and to the reward. The reward was modified to be always greater than or equal to zero, so we can exploit the monotonicity property in order to avoid the policy that always falls getting zero until the end of the episode. So if we use the RCVaR or the Mean-RMAD with $0 \leqslant \lambda \leqslant 0.5$, we have the guarantee that the optimal policy is not the one that commits suicide because all other policies give a reward that is always greater or equal to zero.



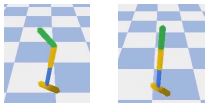
(a) Mean-RMAD frontier



(b) Algorithms comparison

Figure 5: Results obtained on Hopper, with shaded area representing the standard deviation, while the solid lines represent the mean. Figure 5a shows the trade-off between the normalized expected return and the RMAD of the policies trained with the previous algorithms. Figure 5b reports as comparison the learning curves of the newly introduced algorithms and of the baseline TRVO in a risk-averse setting. The lines that connect the points in the frontier are showed only for readability.

Figure 5a shows the trade-off between the normalized expected return and the RMAD. We can see that the algorithms have found risk-neutral policies with high normalized expected return and high RMAD and Volatility; policies that have low normalized expected return and low RMAD and Volatility; other policies that can obtain a good mean even if not high with a quite low RMAD and Volatility. Thanks to the monotonicity property, we obtained that for all values of α that we used (0.9999, 0.8, 0.6, 0.4 and 0.2) and for $0 \leqslant \lambda \leqslant 0.5$ (we used 0, 0.37 and 0.5) the found policies with RCVaR-TRPO, RCVaR-PPO, RMAD-TRPO and RMAD-PPO did never commit suicide. Finally, in Figure 5b we can see that the algorithms have achieved convergence almost at the same time.



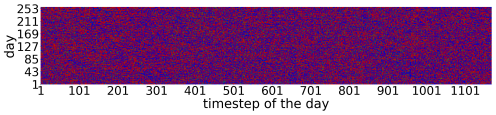
(a) (b)

Figure 6: Frames from the Hopper environment of policies trained with RMAD-TRPO. Figure 6a comes from the risk-neutral policy, while Figure 6b comes by using a risk-aversion factor 0.5. Similar behaviours came with the other algorithms.

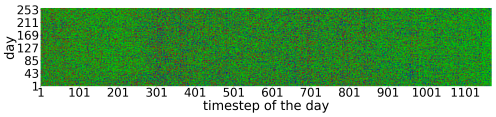
Figure 6a show a frame of the policy risk-neutral obtained from the learning. The policy risk-neutral moves by oscillating the top part of the robot, in this way it can achieve a high normalized expected return but there is a higher risk of falling. While in Figure 6b the frame of a policy risk-averse shows a more cautious behaviour that maintains fixed the top part of the robot, allowing to move forward and reducing the number of falls.

6.3. Trading

The environment consists in a simulated trading task on the Foreign Exchange (FOREX) market. The agent can trade a fixed amount of dollars, based on exchange rate prices taken from real 2017 open data. An episode includes a day of trading from 01:00 to 21:29 with a step of one minute. There are three possible actions: short position (0), flat position (1) and long position (2).



(a) $\lambda = 0$



(b) $\lambda = 0.1$

Figure 7: Actions selected during the year on the Trading environment by the policy trained with RMAD-TRPO and risk-aversion level indicated under each figure. On the x axis there are the timesteps of the day and on the y axis there are the days of the year. The green dots indicate the flat action, the red dots are the short position, while the blue dots are the long position.

In Figure 7 we can see the actions selected by the policies trained with RMAD-TRPO for different risk-aversion levels. The figures with a lot of red and blue dots represent aggressive and risk-neutral policies, while if there are a lot of green dots the figure represents a risk-averse policy. For each algorithm, the risk-averse policies tend to choose the flat action more times than what the risk-neutral policy does, until the most risk-averse policy does nothing and exits from the market. In fact, in Figure 7a we can see that the risk neutral policies trade a lot on the market. While in Figure 7b the policy is risk-averse and chooses several times the flat action. We obtained also that the most risk-averse policy chooses always the flat action. These results indicate that the algorithms are risk sensitive, they are able to find aggressive policies that maximize the mean, policies that give a good mean but with lower risk depending on the risk aversion level and policies that don't want any risk at all. Similar results were obtained also with the other algorithms.

7. Related work

Alexander et al. (2014) illustrates risk-averse optimization and analyzes the coherence properties of several risk measures: utility model, CVaR, VaR, entropic risk measure, mean-variance, mean-deviation and mean-semideviation, which have been used mostly in the form based on the return. The first work about a reward-based risk measure, the Mean-Volatility, is Bisi et al. (2020), but it is not coherent. In Bisi (2022), the author develops the algorithm ROSA, in which you can use any RL method to optimize some return-based risk measures, but at the cost of a greater number of samples due to the augmentation of the state space, while the RCVaR doesn't need it thanks to its reward-based nature. An algorithm similar to RMAD-TRPO and RCVaR-TRPO is TRVO (Bisi et al., 2020), that merged together two streams: risk-averse objective functions and safe policy updates. The first one reduces the inherent risk and the second one reduces the model risk. RMAD-TRPO and RCVaR-TRPO add also the coherence, which provides rational solutions. In Tamar et al. (2015) the authors develop policy gradient for coherent risk measures, but it doesn't provide safe updates. The famous risk-neutral algorithm TRPO, that provides safe updates, was introduced in Schulman et al. (2015). An approximation of TRPO is Proximal Policy Optimization (PPO) of Schulman et al. (2017), both algorithms are used to deal with complex control problems.

8. Conclusions

In this work, we tackled the risk-aversion problem with two new risk measures that are both coherent and based on the reward, which allow to smooth the trajectories avoiding shocks and they allow to obtain solutions that are rational. These measures bound the corresponding return-based measures. We obtained trust-region algorithms for both measures that guarantee safe improvement updates. For the RCVaR we can also apply any classic risk-neutral Reinforcement Learning algorithm maintaining the monotonic improvement. We showed the risk-sensitivity on challenging environments like Hopper and Trading, where we obtained similar convergence speed to that of TRVO. Future studies can further develop the theoretical part of these algorithms: whether they can achieve the global optimal policy or an epsilon optimal policy and the convergence rate. To conclude, we obtained safe methods that share the learning speed of state-of-the-art risk-neutral algorithms while taking into consideration the risk and having the guarantee that the found solutions have good properties thanks to the coherence.

References

S. Alexander, D. Darinka, and R. Andrzej. *Lectures on Stochastic Programming: Modeling and Theory*. SIAM - Society for Industrial and Applied Mathematics, 2014.

L. Bisi. *Algorithms for risk-averse reinforcement learning*. PhD thesis, Politecnico di Milano, 2022.

L. Bisi, L. Sabbioni, E. Vittori, M. Papini, and M. Restelli. Risk-Averse Trust Region Optimization for Reward-Volatility Reduction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4583–4589, 2020.

J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel. Trust region policy optimization. *CoRR*, abs/1502.05477, 2015.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

A. Tamar, Y. Chow, M. Ghavamzadeh, and S. Mannor. Policy gradient for coherent risk measures. *CoRR*, abs/1502.03919, 2015.