

Risk-averse Distributional Reinforcement Learning

A CVaR optimization approach

Silvestr Stanko¹

¹Department of Computer Science
Czech Technical University

February 5, 2018

Outline

- 1 Introduction
 - Motivation
 - Conditional Value-at-Risk
- 2 CVaR Value Iteration
 - Previous results
 - Linear-time improvement
- 3 Other Results

Motivation



Figure: Robotics



Figure: Finance

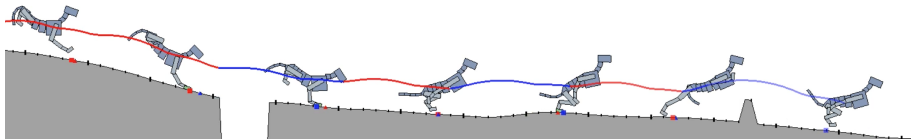


Figure: Reinforcement Learning safety

Value-at-Risk

- facts
- more facts

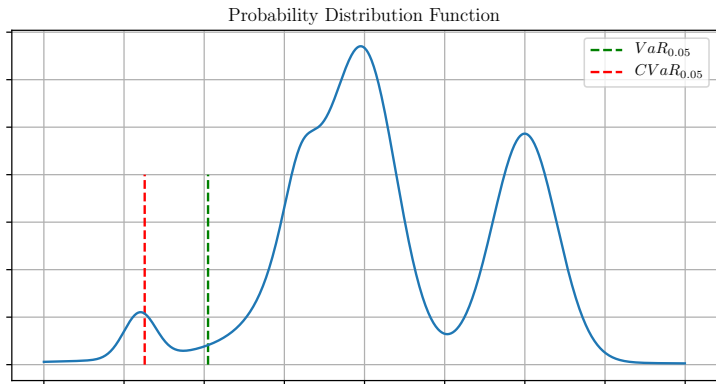
$$\text{VaR}_\alpha(Z) = F^{-1}(\alpha) = \max \{z | F(z) \leq \alpha\}$$

Conditional Value-at-Risk

- facts
- more facts

$$\text{CVaR}_\alpha(Z) = \frac{1}{\alpha} \int_0^\alpha F_Z^{-1}(\beta) d\beta = \frac{1}{\alpha} \int_0^\alpha \text{VaR}_\beta(Z) d\beta$$

VaR, CVaR



Reinforcement Learning

Definition

An MDP is a 5-tuple $\mathcal{M} = (\mathcal{X}, \mathcal{A}, R, P, \gamma)$, where \mathcal{X} and \mathcal{A} are the finite state and action spaces.

$R(x, a) \in [R_{\min}, R_{\max}]$ is a random variable representing the reward generated by being in state x and selecting action a ; $P(\cdot|x, a)$ is the transition probability distribution; $\gamma \in [0, 1)$ is a discount factor.

We also assume we are given a starting state x_0 .

Goal

Definition

$Z^\pi(x_t)$ Is a random variable representing the discounted reward along a trajectory generated by the MDP by following the policy π , starting at state x_t .

$$Z^\pi(x_t) = \sum_{t=0}^{\infty} \gamma^t R(x_t, \pi(x_t))$$

Reinforcement Learning with CVaR

For a given α , our goal is to find a globally optimal policy π^*

$$\pi^* = \arg \max_{\pi} CVaR_{\alpha}^{\pi}(Z^{\pi}(x_0))$$

Value Iteration

Theorem (CVaR decomposition)

For any $t \geq 0$, denote by $Z = (Z_{t+1}, Z_{t+2}, \dots)$ the reward sequence from time $t + 1$ onwards. The conditional CVaR under policy π obeys the following decomposition:

$$CVaR_{\alpha}(Z^{\pi}(x, a)) = \min_{\xi \in \mathcal{U}_{CVaR}(\alpha, P(\cdot|x, a))} \sum_{x'} p(x'|x, a) \xi(x') CVaR_{\xi(x')\alpha}(Z^{\pi}(x'))$$

Theorem (CVaR Value Iteration)

The following Bellman operator is a contraction:

$$\mathbf{T}V(x, y) = \max_a \left[R(x, a) + \gamma \min_{\xi} \sum_{x'} p(x'|x, a) \xi(x') V(x', y\xi(x')) \right]$$

CVaR Value Iteration

Theorem (CVaR Value Iteration)

The following Bellman operator is a contraction:

$$\mathbf{T}V(x, y) = \max_a \left[R(x, a) + \gamma \min_{\xi} \sum_{x'} p(x'|x, a) \xi(x') V(x', y\xi(x')) \right]$$

The operator \mathbf{T} describes the following relationship:

$$\mathbf{T}CVaR_y(Z(x)) = \max_a [R(x, a) + \gamma CVaR_y(Z(x, a))]$$

Linear interpolation

Computing operator \mathbf{T} is intractable, as the state-space is continuous. A solution would be to approximate the operator with linear interpolation.

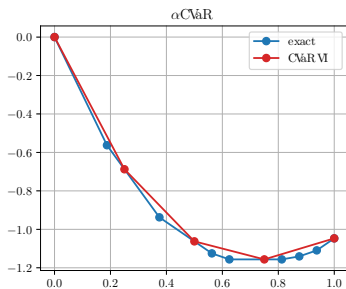
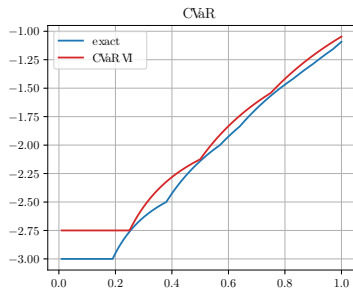
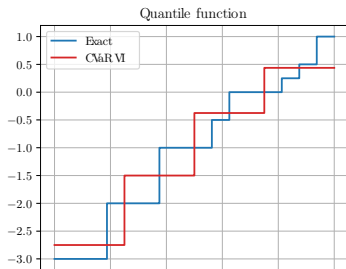
Theorem

The function $\alpha CVaR_\alpha$ is convex. The operator $\mathbf{T}_\mathcal{I}$ is a contraction.

$$\mathcal{I}_x[V](y) = y_i V(x, y_i) + \frac{y_{i+1} V(x, y_{i+1}) - y_i V(x, y_i)}{y_{i+1} - y_i} (y - y_i)$$

$$\mathbf{T}_\mathcal{I} V(x, y) = \max_a \left[R(x, a) + \gamma \min_\xi \sum_{x'} p(x'|x, a) \frac{\mathcal{I}_{x'}[V](y\xi(x'))}{y} \right]$$

This iteration can be formulated and solved as a linear program.



αCVaR_α duality

Lemma

Any discrete distribution has a piecewise linear αCVaR_α function. Similarly, any a piecewise linear αCVaR_α function can be seen as representing a certain discrete distribution.

$$\alpha\text{CVaR}_\alpha \Leftarrow \text{VaR}$$

$$\frac{\partial}{\partial \alpha} \alpha\text{CVaR}_\alpha(Z) = \frac{\partial}{\partial \alpha} \int_0^\alpha \text{VaR}_\beta(Z) d\beta = \text{VaR}_\alpha(Z)$$

$$\alpha\text{CVaR}_\alpha \Rightarrow \text{VaR}$$

$$\alpha\text{CVaR}_\alpha(Z) = \int_0^\alpha \text{VaR}_\beta(Z) d\beta$$

Linear-time Computation

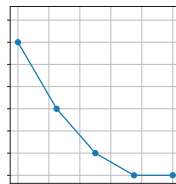
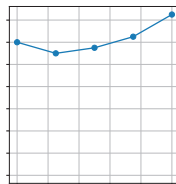
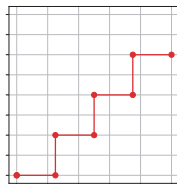
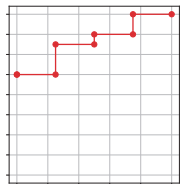
Theorem

Solution to minimization problem present in the CVaR Value Iteration can be computed by setting

$$\xi(x') = \frac{F_{x'}(F_x^{-1}(\alpha))}{\alpha}$$

The computational complexity is $O(n \cdot m)$ where n is the number of transition states and m is the number of atoms.

Next state CVaR computation

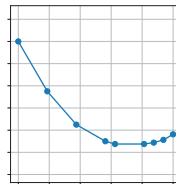
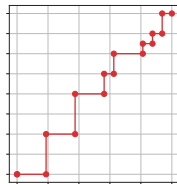


0.25

0.75

0.25

0.75



VaR-based Policy Improvement

Theorem

Let π be a fixed policy, $\alpha \in (0, 1]$. By following policy π' from the following algorithm, we will improve $CVaR_\alpha(Z)$ in expectation:

$$CVaR_\alpha(Z^\pi) \leq CVaR_\alpha(Z^{\pi'})$$

```

input  $\alpha, x_0, \gamma$ 
 $a = \arg \max_a CVaR_\alpha(Z(x_0, a))$ 
 $s = VaR_\alpha(Z(x_0, a))$ 
 $x_t, r_t = \text{envTransition}(x_0, a)$ 
while  $x_t$  is not terminal do
     $s = \frac{s - r_t}{\gamma}$ 
     $a = \arg \max_a \mathbb{E}[(Z(x_t, a) - s)^-]$ 
     $x_t, r_t = \text{envTransition}(x_t, a)$ 
end while
  
```


Implementations

`https://github.com/Silvicek/policy-improvement`

`https://github.com/Silvicek/distributional-dqn`

TODO

- CVaR Q-learning
 - (?) Use Wasserstein distance with quantile improvement
 - (?) Extend the VaR-based algorithm
 - (?) Combine with quantile regression
- Experiments
 - Value Iteration + Q-learning
 - Deep Q-learning