Risk-averse Distributional Reinforcement Learning

A CVaR optimization approach

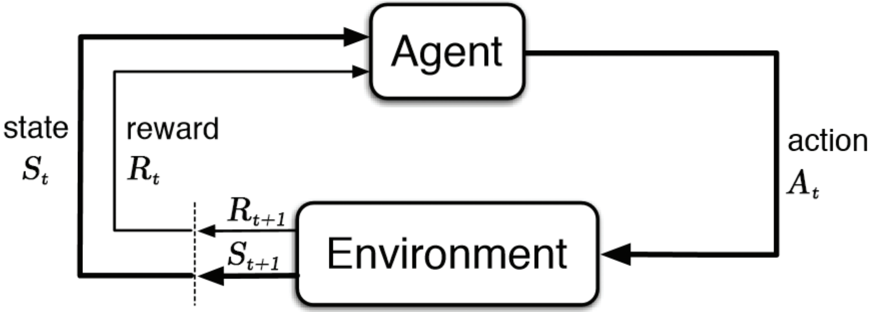
Silvestr Stanko[1]
Supervisor: Karel Macek, Ph.D.[2]

[1]Department of Computer Science
Czech Technical University

[2]DHL Information Services

Monday 11th June, 2018

Reinforcement Learning



Reinforcement Learning goals

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(x_t, \pi(x_t))\right]$$

Risk-averse Reinforcement Learning: Motivation

Example

You have to deliver a package fast. Choose between:

1. Shorter route with the risk of running into traffic
2. Longer but safe route

Figure: AI Safety

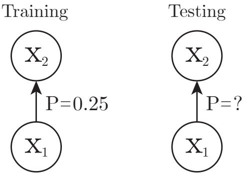
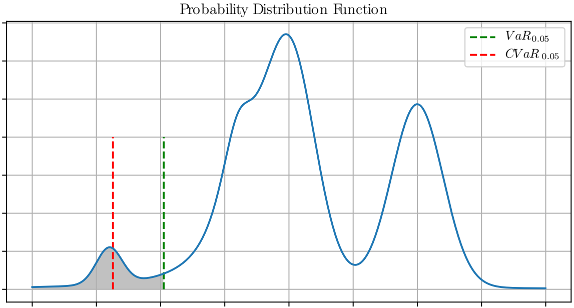


Figure: Robustness

Figure: Critical Applications

Value-at-Risk, Conditional Value-at-Risk



Reinforcement Learning with CVaR

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

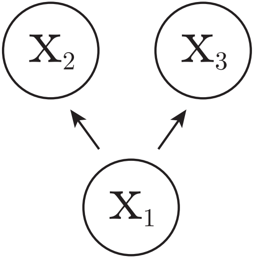
$$\pi^* = \arg\max_{\pi} \mathrm{CVaR}_\alpha(Z^\pi(x_0))$$

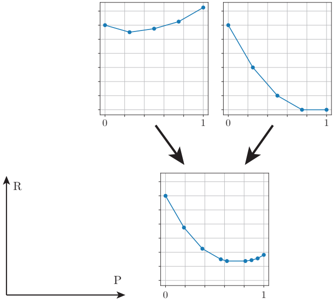
CVaR Value Iteration

$C(x, \alpha)$ represents CVaR_α when following the optimal CVaR policy

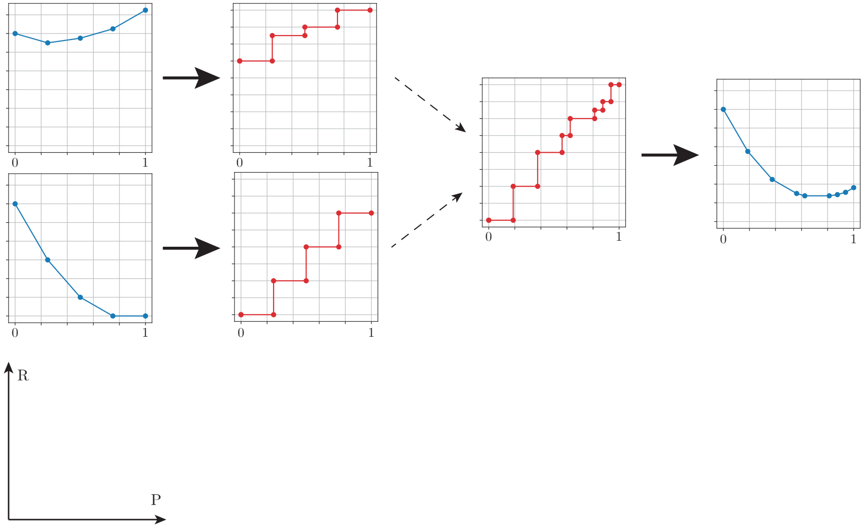
CVaR Value Iteration

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





Bonus: CVaR computation via quantile representation



Linear-time Computation

Theorem

Solution to minimization problem

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

can be computed by setting

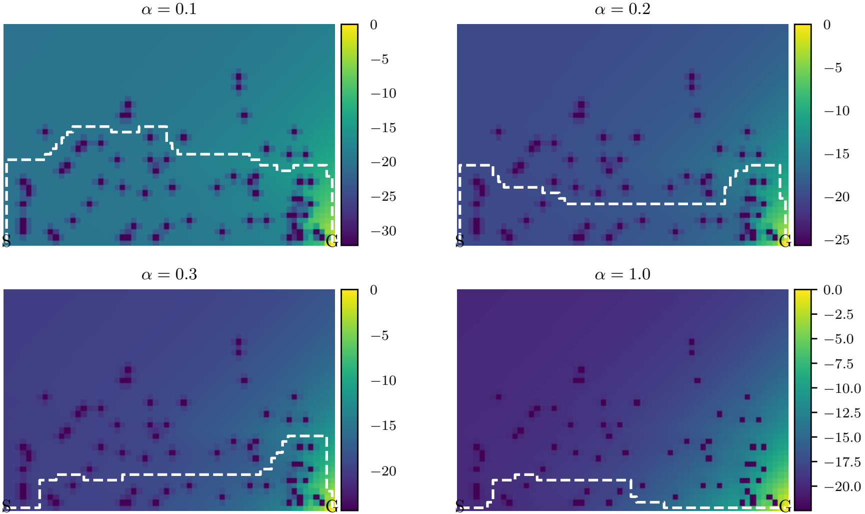
$$\xi(x') = \frac{F_{Z(x')}(F^{-1}_{Z(x,a)}(\alpha))}{\alpha}$$

The computational complexity improvement is

$$O(m^{4.5}n^2) \to O(mn)$$

where m is the number of transition states and n is the number of atoms.

CVaR Value Iteration - Experiments



CVaR Q-learning

Pseudocode

1. Sample a transition x, a, x', r
2. Create a target distribution $\mathbf{d}$
3. Update current estimates of VaR and CVaR proportionally to the target distribution

Recursive CVaR Estimation

$$V_{t+1} = V_t + \beta_t \left[1 - \frac{1}{\alpha} \mathbb{1}_{(V_t \geq r)} \right]$$

$$C_{t+1} = (1 - \beta_t) C_t + \beta_t \left[V_t + \frac{1}{\alpha} (r - V_t)^- \right]$$

VaR-based Policy Improvement

Theorem

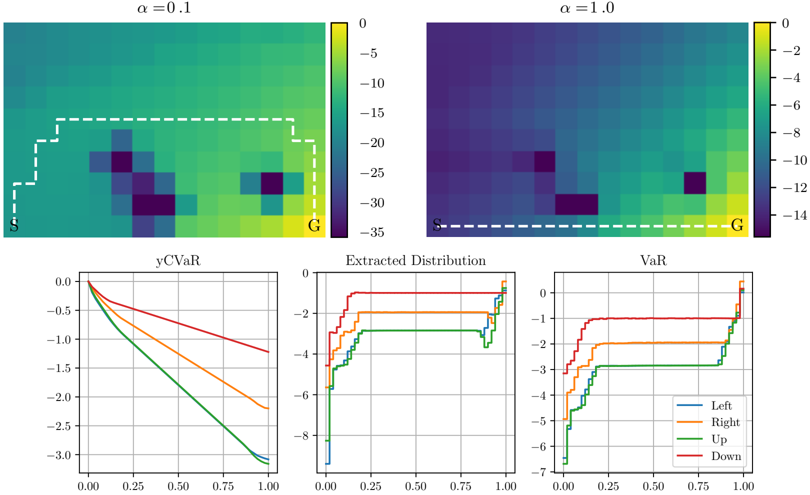
Let π be a fixed policy, $\alpha \in (0, 1]$. By following policy π' from the following algorithm, we will improve $CVaR_\alpha(Z)$ in expectation:

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

VaR-based Policy Improvement

$a = \arg\max_a \mathrm{CVaR}_\alpha(Z(x_0, a))$
$s = \mathrm{VaR}_\alpha(Z(x_0, a))$
Take action a, observe x, r
while x is not terminal **do**
  $s = \dfrac{s - r}{\gamma}$
  $a = \arg\max_a \mathbb{E}\left[(Z(x, a) - s)^-\right]$
  Take action a, observe x, r
end while

CVaR Q-learning - Experiments



Approximate Q-learning

Problem

Q-learning is intractable for large state spaces.

Solution

Use approximate Q-learning:

1. Formulate CVaR Q-learning update as a minimizing argument
2. Use methods of convex optimization to find the optimal point

TD update $\rightarrow$ loss function

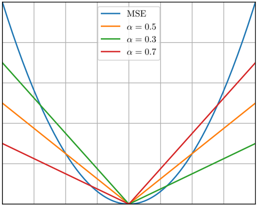
VaR loss

$$\mathcal{L}_{\text{VaR}} = \sum_{i=1}^{N} \mathbb{E}_j \left[(r + \gamma d_j - V_i(x,a))(y_j - \mathbb{1}_{(V_i(x,a) \geq r + \gamma d_j)}) \right]$$

CVaR loss

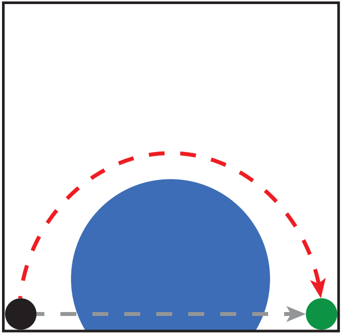
$$\mathcal{L}_{\text{CVaR}} = \sum_{i=1}^{N} \mathbb{E}_j \left[\left(V_i(x,a) + \frac{1}{y_i} (r + \gamma d_j - V_i(x,a))^{-} - C_i(x,a) \right)^2 \right]$$

$$\mathcal{L} = \mathbb{E}\left[\mathcal{L}_{\text{VaR}} + \mathcal{L}_{\text{CVaR}}\right]$$



Deep CVaR Q-learning - Experiments

- Model: Convolutional Neural Network
- Environment: Ice Lake



1. Video: $\alpha = 1$
2. Video: $\alpha = 0.3$

Summary

1. **Faster CVaR Value Iteration**
   - $O(m^{4.5}n^2) \to O(mn)$
   - Formally proved for increasing, unbounded distributions.
   - Experimentally verified for general distributions.
2. **CVaR Q-learning**
   - Sampling version of CVaR Value Iteration.
   - Based on the distributional approach.
   - Experimentally verified.
3. **Distributional Policy improvement**
   - Proved monotonic improvement for distributional RL.
   - Used as a heuristic for extracting π^* from CVaR Q-learning.
4. **Deep CVaR Q-learning**
   - TD update $\to$ loss function.
   - Experimentally verified in a deep learning context.