Risk-Sensitive Reward-Free Reinforcement Learning with CVaR

Xinyi Ni, Guanlin Liu, Lifeng Lai

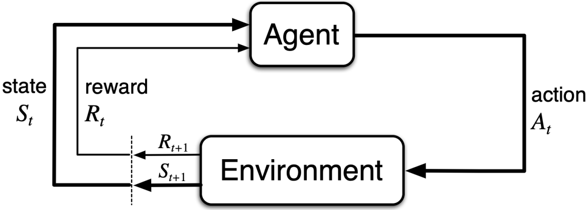
University of California, Davis

ICML, 2024

Overview

Outline

Reinforcement Learning (RL)



- Agent and environment interact at discrete time $t = 0, 1, 2, 3, \ldots$
- At each time step t, agent observes the state S_t
- take action A_t
- get reward R_t
- go to the corresponding next state S_{t+1}

Limitations of Existing Exploration Algorithms

- Simple exploration methods, which can be inefficient for discovering high-reward states, may result in high sample complexity. Sophisticated exploration strategies that are efficient for prespecified reward function exist.

Limitations of Existing Exploration Algorithms

- Simple exploration methods, which can be inefficient for discovering high-reward states, may result in high sample complexity. Sophisticated exploration strategies that are efficient for prespecified reward function exist.
- In practice, reward functions are typically iteratively engineered to encourage desired behavior. If sophisticated methods are applied for each time reward functions are updated, it can be sample inefficient.

Reward-Free RL (Jin et al., 2020)

- Goal: develop a more efficient exploration approach that doesn't rely on explicit reward information.

Reward-Free RL (Jin et al., 2020)

- Goal: develop a more efficient exploration approach that doesn't rely on explicit reward information.
- Exploration Phase: efficiently explore the environment without reward information through Protocol 1.

Protocol 1 Reward-Free Exploration

for $k = 1$ **to** K **do**
  learner decides a policy π_k
  environment samples the initial state $s_0 \sim \mathbb{P}_1$.
  for $h = 1$ **to** H **do**
    learner selects action $a_h \sim \pi_h(\cdot|s_h)$
    environment transitions to $s_{h+1} \sim \mathbb{P}_h(\cdot|s_h, a_h)$
    learner observes the next state s_{h+1}

Reward-Free RL (Jin et al., 2020)

- Goal: develop a more efficient exploration approach that doesn't rely on explicit reward information.
- Exploration Phase: efficiently explore the environment without reward information through Protocol 1.

Protocol 1 Reward-Free Exploration

```
for k = 1 to K do
    learner decides a policy π_k
    environment samples the initial state s_0 ~ P_1.
    for h = 1 to H do
        learner selects action a_h ~ π_h(·|s_h)
        environment transitions to s_{h+1} ~ P_h(·|s_h, a_h)
        learner observes the next state s_{h+1}
```

- Planning Phase: derive a near-optimal policy with any given reward only using the dataset gathered during the exploration phase without further interaction with the environment.

Risk-Sensitive RL

- In safety-critical scenarios, decision-makers prioritize mitigating low-probability but high-impact risks.

Risk-Sensitive RL

- In safety-critical scenarios, decision-makers prioritize mitigating low-probability but high-impact risks.
- Many risk measures have been investigated, but coherent risk measures are preferred due to their properties: 1) monotonicity; 2) positive-homogeneity; 3) sub-additivity; 4) translation-invariance.

Conditional Value-at-Risk (CVaR)

- For a random variable X, CVaR at a given risk tolerance $\tau \in (0,1]$ is defined as

$$\mathrm{CVaR}_{\tau}(X) := \sup_{b \in \mathbb{R}} \left(b - \tau^{-1} \mathbb{E}[(b - X)^{+}] \right), \tag{1}$$

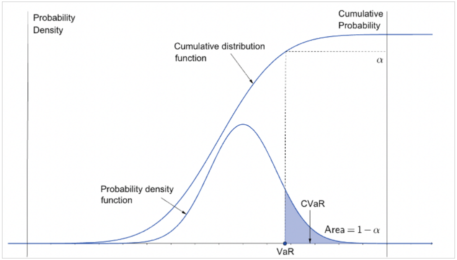
where $x^{+} := \max(0, x)$.

Conditional Value-at-Risk (CVaR)

- For a random variable X, CVaR at a given risk tolerance $\tau \in (0, 1]$ is defined as

$$\mathrm{CVaR}_\tau(X) := \sup_{b \in \mathbb{R}} \left(b - \tau^{-1}\mathbb{E}[(b - X)^+]\right), \tag{1}$$

  where $x^+ := \max(0, x)$.
- CVaR is coherent and can effectively quantify the average outcome in the worst τ-percentile of scenario.



Outline

Motivation

- Is it possible to design provably efficient risk-sensitive reward-free RL algorithm?

Motivation

- Is it possible to design provably efficient risk-sensitive reward-free RL algorithm?
- CVaR-RF-Exploration: design an efficient CVaR reward-free exploration algorithm.
- CVaR-RF-Planning: derive a PAC algorithm to solve CVaR RL with any given reward function based on the dataset gathered in exploration phase.

Motivation

- Is it possible to design provably efficient risk-sensitive reward-free RL algorithm?
- CVaR-RF-Exploration: design an efficient CVaR reward-free exploration algorithm.
- CVaR-RF-Planning: derive a PAC algorithm to solve CVaR RL with any given reward function based on the dataset gathered in exploration phase.

Definition

A CVaR-RF exploration algorithm is (ϵ, δ)-PAC with a given risk tolerance τ if for any reward function r,

$$\mathbb{P}\left(\mathbb{E}_{s_1 \sim \mathbb{P}_1}\left[\mathrm{CVaR}_\tau^\star(s_1; r) - \mathrm{CVaR}_\tau^{\hat{\rho}}(s_1; r)\right] \leq \epsilon\right) \geq 1 - \delta.$$

Augmented MDP

- (Baüerle & Ott, 2011) establish the existence of an optimal policy that is deterministic and Markovian within the augmented MDP (Π^{Aug}). The augmented state space is denoted by $\mathcal{S}^{\text{Aug}} = \mathcal{S} \times [0, H]$, where $[0, H]$ is the augmented space for initial budget b.

Value Functions

- For any policy $\rho \in \Pi^{\mathrm{Aug}}$, we define:

$$V_h^{\rho}(s_h, b_h) = \mathbb{E}_{\rho}\left[\left(b_h - \sum_{h'=h}^{H} r_{h'}(s_{h'}, a_{h'})\right)^{+} \middle| s_h, b_h\right],$$

$$Q_h^{\rho}(s_h, b_h, a_h) = \mathbb{E}_{\rho}\left[\left(b_h - \sum_{h'=h}^{H} r_{h'}(s_{h'}, a_{h'})\right)^{+} \middle| s_h, b_h, a_h\right].$$

Objective and Goal

- The CVaR objective is

$$\mathrm{CVaR}_\tau^\rho(s_1) = \max_{b_1 \in [0,H]} \{ b_1 - \tau^{-1} V_1^\rho(s_1, b_1) \}.$$

- The goal is to optimize

$$\mathrm{CVaR}_\tau^\star(s_1) = \max_{b_1 \in [0,H]} \{ b_1 - \tau^{-1} \min_{\rho \in \Pi^{\mathrm{Aug}}} V_1^\rho(s_1, b_1) \},$$

and find the corresponding optimal policy $\rho^\star$ with optimal initial budget $b_1^\star$.

Bellman Equations

- For CVaR RL, the Bellman Equations are defined as:

$$V_h^\rho(s_h, b_h) = \mathbb{E}_{a_h \sim \rho_h(s_h, b_h)} \left[Q_h^\rho(s_h, b_h, a_h) \right],$$
$$Q_h^\rho(s_h, b_h, a_h) = \left[\mathbb{P}_h V_{h+1} \right](s_h, b_h, a_h),$$

where $b_{h+1} = b_h - r_h$ and $V_{H+1}^\rho(s, b) = b_1^+ := \max(0, b_1)$.

Bellman Equations

- For CVaR RL, the Bellman Equations are defined as:

$$
\begin{aligned}
V_h^\rho(s_h, b_h) &= \mathbb{E}_{a_h \sim \rho_h(s_h, b_h)} \left[Q_h^\rho(s_h, b_h, a_h) \right], \\
Q_h^\rho(s_h, b_h, a_h) &= \left[\mathbb{P}_h V_{h+1} \right](s_h, b_h, a_h),
\end{aligned}
$$

where $b_{h+1} = b_h - r_h$ and $V_{H+1}^\rho(s, b) = b_1^+ := \max(0, b_1)$.

- Similarly, we define the optimal conditions as:

$$
\begin{aligned}
V_h^\star(s_h, b_h) &= \min_{a \in \mathcal{A}} Q_h^\star(s_h, a_h, b_h), \\
\rho_h^\star(s_h, b_h) &= \text{argmin}_{a \in \mathcal{A}} Q_h^\star(s_h, b_h, a_h)], \\
Q_h^\star(s_h, b_h, a_h) &= \left[\mathbb{P}_h V_{h+1}^\star \right](s_h, b_h, a_h),
\end{aligned}
$$

where $b_{h+1} = b_h - r_h$ and $V_{H+1}^\star(s, b) = b_1^+ = \max(0, b_1)$.

- The optimality has been demonstrated in (Wang et al., 2023).

Outline

Key Lemma

Lemma

An algorithm is (ϵ, δ)-PAC for CVaR-RF exploration with a given risk tolerance τ if for any reward function r and for any $b_1 \in [0, H]$,

$$\left| V_1^{\rho}(s_1, b_1; r) - \hat{V}_1^{\rho}(s_1, b_1; r) \right| \leq \epsilon\tau/3.$$

Key Lemma

Lemma

An algorithm is (ϵ, δ)-PAC for CVaR-RF exploration with a given risk tolerance τ if for any reward function r and for any $b_1 \in [0, H]$,
$$\left| V_1^{\rho}(s_1, b_1; r) - \hat{V}_1^{\rho}(s_1, b_1; r) \right| \leq \epsilon\tau/3.$$

- Establish a connection between CVaR-RF RL with risk-neutral reward-free RL and solve the complexity added by the adoption of CVaR.

Methodology

- Assume the optimization error in planning phase is bounded (could be easily satisfied by existing CVaR RL algorithms).

Methodology

- Assume the optimization error in planning phase is bounded (could be easily satisfied by existing CVaR RL algorithms).
- Define the estimation error:

$$\hat{e}_h^{t,\rho}(s_h, b_h, a_h; r) := \left| \hat{Q}_h^{t,\rho}(s_h, b_h, a_h; r) - Q_h^{\rho}(s_h, b_h, a_h; r) \right|.$$

Error Upper Bound

Definition

The upper confidence bound $E_h^t(s_h, a_h)$ for the error, recursively defined as follows: $E_{H+1}^t(s, a) = 0$ for all (s, a), and for all $h \in [H]$, with the convention $\frac{1}{0} = +\infty$,

$$E_h^t(s_h, a_h) = \min\left\{H, H\sqrt{\frac{2\beta(n_h^t(s,a),\delta)}{n_h^t(s,a)}} + \sum_{s'} \hat{\mathbb{P}}_h^t(s'|s,a) \max_a E_{h+1}^t(s', a)\right\},$$

(refers to Eq (8) in Algorithm 1)

where $\beta(n, \delta)$ is a threshold function, an input to the algorithm, the choice of which will be discussed later.

Error Upper Bound

- Consider an event

$$\mathcal{E} = \Big\{ \forall t \in \mathbb{N}, \forall h \in [H], \forall (s,a), \\ \mathrm{KL}(\hat{\mathbb{P}}_h^t(\cdot|s,a), \mathbb{P}^h(\cdot|s,a)) \le \frac{\beta(n_h^t(s,a),\delta)}{n_h^t(s,a)} \Big\},$$

Error Upper Bound

- Consider an event

$$\mathcal{E} = \Big\{ \forall t \in \mathbb{N}, \forall h \in [H], \forall (s,a),$$
$$\mathrm{KL}(\hat{\mathbb{P}}_h^t(\cdot|s,a), \mathbb{P}^h(\cdot|s,a)) \leq \frac{\beta(n_h^t(s,a),\delta)}{n_h^t(s,a)} \Big\},$$

Theorem

For any policy ρ, any reward function r and any b,

$$\hat{e}_h^{t,\rho}(s,b,a;r) \leq E_h^t(s,a)$$

holds on event $\mathcal{E}$.

Algorithm Design

- **Sampling rule:** the exploration policy π^{t+1} is the greedy policy with respect to $E^t(s,a)$, such that for all $s \in \mathcal{S}$ and $h \in [H]$:

$$\pi_h^{t+1}(s_h) = \mathrm{argmax}_a E_h^t(s,a). \quad \text{(refers to Eq (9) in Algorithm 1)}$$

- **Stopping rule:** the algorithm stops at

$$t_{\mathrm{stop}} = \inf\{t : E_h^t(s_1, \pi_1^{t+1}(s_1)) \leq \epsilon\tau/3\}.$$

Algorithm 1 CVaR-RF-UCRL

```
Given: risk tolerance τ ∈ (0,1]
Initialization: t = 1, D_0 = ∅, initialize E^0 with (8) and π_h^1 with (9)
while E_h^{t-1}(s_1, π_1^t(s_1)) ≥ ετ/3 do
    Observe the initial state s_1^t ~ P_0
    for h = 1, ..., H-1, H do
        Play a_h^t ~ π_h^t(s_h^t)
        Observe the next state s_{h+1}
    end for
    Compute E^t according to (8) and π^{t+1} according to (9)
    D_t = D_{t-1} ∪ (s_1^t, a_1^t, ..., s_H^t, a_H^t)
    t = t + 1
end while
Return the dataset D_{t_stop}
```

Theoretical Guarantees

Theorem

Using threshold $\beta(n, \delta) = \log(2SAH/\delta) + (S-1)\log(e(1 + n/(S-1)))$*, the CVaR-RF-UCRL is* (ϵ, δ)*-PAC for CVaR-RF exploration. The number of trajectories collected in the exploration phase is bounded by* $\tilde{\mathcal{O}}\left(\frac{S^2AH^4}{\epsilon^2\tau^2}\right)$.

Theoretical Guarantees

Theorem

Using threshold $\beta(n, \delta) = \log(2SAH/\delta) + (S-1)\log(e(1 + n/(S-1)))$, *the CVaR-RF-UCRL is* (ϵ, δ)*-PAC for CVaR-RF exploration. The number of trajectories collected in the exploration phase is bounded by* $\tilde{\mathcal{O}}\left(\frac{S^2AH^4}{\epsilon^2\tau^2}\right)$.

- Compared with risk-neutral reward-free approaches, the denominator of the bound we obtained is related to the risk tolerance parameter τ.
- This is expected since CVaR is interpreted as the mean of the tail distribution with an area under the curve equal to τ, it requires more trajectories for smaller τ values and fewer trajectories for larger τ values.

Outline

Framework

- Compute the empirical transition matrix based on the dataset collected by CVaR-RF-UCRL
- Find a near-optimal policy by employing a 'APPROXIMATE-CVaR-SOLVER', which can be any algorithm designed to find an $\delta/3$-suboptimal policy for CVaR RL with known transition matrix and reward.

Algorithm 2 CVaR-RF-Planning

1: **Input:** a dataset of transition $\mathcal{D}_{t_{\text{stop}}}$, reward function r, accuracy ϵ, risk tolerance τ.
2: **for** all $(s, a, s', h) \in \mathcal{S} \times \mathcal{A} \times \mathcal{S} \times [H]$ **do**
3: $\quad N_h(s, a, s') \leftarrow \sum_{(s_h, a_h, s_{h+1}) \in \mathcal{D}} \mathbb{I}[s_h = s, a_h = a, s_{h+1} = s']$.
4: $\quad N_h(s, a) \leftarrow \sum_{s'} N_h(s, a, s')$.
5: $\quad \hat{\mathbb{P}}_h(s'|s, a) = N_h(s, a, s')/N_h(s, a)$.
6: **end for**
7: $\hat{\rho}, \hat{b} \leftarrow$ APPROXIMATE-CVaR-SOLVER$(\hat{\mathbb{P}}, r, \epsilon, \tau)$.
8: return policy $\hat{\rho}$, and initial budget $\hat{b}$.

Approximate-CVaR-Solver

- Iteratively solve the Bellman optimality equations in a dynamic programming manner.
- The greedy policy induced by the resulting $Q^\star$ yields the optimal policy without errors.

Algorithm 3 CVaR-VI

Input: transition matrix $\mathbb{P}$, reward function r, risk tolerance τ

for all $s \in \mathcal{S}, b \in [0, H]$ **do**

  Set $V_{H+1}(s, b) = b^+$

  for $h = H, H-1, \ldots, 1$ **do**

    $Q_h(s_h, b_h, a_h) = [\mathbb{P}_h V_{h+1}](s_h, b_h, a_h)$, where $b_{h+1} = b_h - r_h$

    $\rho_h^\star(s_h, b_h) = \mathrm{argmin}_a Q_h(s_h, b_h, a_h)$

    $V_h^\star(s_h, b_h) = \min_a Q_h(s_h, b_h, a_h)$

  end for

end for

Calculate $b^\star = \mathrm{argmax}_{b_1 \in [0,1]} \left\{ b - \tau^{-1} V_1(s_1, b) \right\}$

return policy $\rho^\star$ and $b^\star$

Discretization

- CVaR-VI faces computational challenges due to the dynamic programming step, which requires optimization over all $b \in [0, H]$, involving the maximization of a non-concave function.

Algorithm 4 CVaR-VI-DISC

Input: transition matrix $\mathbb{P}$, reward function r, precision parameter η, risk tolerance τ.

Discretize the reward funtion r by

$$\hat{r} = \phi(r) = \eta \lceil r/\eta \rceil \wedge 1$$

for all $s \in \mathcal{S}, \hat{b} = n \cdot \eta, n = 0, 1, \ldots$ **do**

  Set $\hat{V}_{H+1}(s, \hat{b}) = \hat{b}^+$

  for $h = H, H-1, \ldots, 1$ **do**

    $\hat{Q}_h(s_h, \hat{b}_h, a_h) = \left[\mathbb{P}_h \hat{V}_{h+1}\right](s_h, \hat{b}_h, a_h)$, where $\hat{b}_{h+1} = \hat{b}_h - \hat{r}_h$

    $\hat{\rho}^\star_h(s_h, \hat{b}_h) = \mathrm{argmin}_a \hat{Q}_h(s_h, \hat{b}_h, a_h)$

    $\hat{V}^\star_h(s_h, \hat{b}_h) = \min_a \hat{Q}_h(s_h, \hat{b}_h, a_h)$

  end for

end for

Calculate $\hat{b}^\star = \mathrm{argmax}_{\hat{b}} \left\{\hat{b} - \tau^{-1}\hat{V}_1(s_1, \hat{b})\right\}$

return policy $\hat{\rho}^\star$ and $\hat{b}^\star$

Computational Complexity and Error

Theorem

The CVaR-VI-DISC has a run time of $\mathcal{O}(S^2 AH\eta^{-2})$ *in the discretized MDP. Setting* $\eta = \epsilon\tau/3H$*, the run time is* $\mathcal{O}(\frac{S^2AH^3}{\epsilon^2\tau^2})$.

Theorem

By selecting $\eta \leq \epsilon\tau/3H$*, we ensure that*

$$|CVaR_{\tau}^{\rho^\star}(s_1; r) - CVaR_{\tau}^{\hat{\rho}}(s_1; r)| \leq \epsilon/3, \tag{2}$$

where $\rho^\star$ *represents the policy generated by Algorithm 3 and* $\hat{\rho}$ *is the output of Algorithm 4. Consequently, the optimization error is bounded by* $\epsilon/3$*, which satisfies the assumption.*

Adaptability

Proposition

For any $\tau' \geq \tau$, the exploration dataset obtained through Algorithm 1 at risk tolerance τ contains the requisite information for conducting CVaR-RF RL with any higher risk tolerance τ'. Consequently, the planning phase is also compatible with any given $\tau' \geq \tau$.

- Underscore the adaptability of our exploration process to different levels of risk tolerance τ:

Lower Bound

Theorem

Consider a universal constant $C > 0$. For a given risk tolerance $\tau \in (0, 1]$, if the number of actions $A \geq 2$, the number of states $S \geq C \log_2 A + 2$, the horizon $H \geq C \log_2 S + 1$, and the accuracy parameter $\epsilon \leq \min\{1/4\tau, H/48\tau\}$, then any CVaR-RF exploration algorithm that can output ϵ-optimal policies for an arbitrary number of adaptively chosen reward functions with a success probability $\delta = 1/2$ must collect at least $\Omega(S^2AH^2/\tau\epsilon^2)$ trajectories in expectation.

- Compared with the lower bound, the upper bound established before has by an additional factor of H^2 and $1/\tau$, while being tight with respect to the parameters S, A, ϵ. If τ is a constant, our result is nearly minimax-optimal with an additional factor on H^2.

Outline

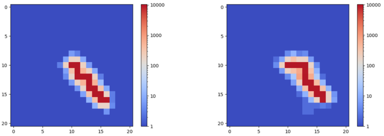
Environment

- A grid-world consisting of 21×21 states, where each state offers four possible actions: up, down, left, right.
- Agent will move to the correct state with a prob of 0.95 and to one of the other three directions with a prob of $0.05/3$ each.

Reward

- Setup 1: The agent starts at position (10,10), and the reward is 0 for most states except at (16,16) ,where it is 1.
- Setup 2: The agent starts at position (10,10), and the reward is 0.5 for most states except at (16,16), where it is 1, and a zero-reward zone marked 'x' from (12,10) to (12,16) (obstacles)

Results

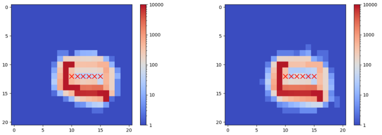


(a) Optimal policy (b) CVaR-VI-DISC

Figure 1. Number of state visits following policies generated under $\mathbb{P}$ and $\hat{\mathbb{P}}$ in reward setup 1 with risk tolerance $\tau = 0.05$.

ϵ, τ	$\mathrm{CVaR}_{\mathbb{P}}$	$\mathrm{CVaR}_{\hat{\mathbb{P}}}$	Error
$0.1, 0.05$	4.308	4.258	0.05
$0.1, 0.95$	4.960	4.954	0.006

Table 1. CVaR values under reward setup 1 with different τ.



(a) Optimal policy (b) CVaR-VI-DISC

Figure 2. Number of state visits following policies generated under $\mathbb{P}$ and $\hat{\mathbb{P}}$ in reward setup 2 with risk tolerance $\tau = 0.05$.

ϵ, τ	$\mathrm{CVaR}_{\mathbb{P}}$	$\mathrm{CVaR}_{\hat{\mathbb{P}}}$	Error
$0.1, 0.05$	1.852	1.829	0.023
$0.1, 0.95$	1.993	1.990	0.003

Table 2. CVaR values under reward setup 2 with different τ.

Outline

Conclusion

- Introduced CVaR-RF, which is able to solve CVaR RL for given any reward function after a singular reward-free exploration.

Conclusion

- Introduced CVaR-RF, which is able to solve CVaR RL for given any reward function after a singular reward-free exploration.
- Proposed CVaR-RF-UCRL as the exploration algorithm and established upper and lower bounds for the sample complexity.

Conclusion

- Introduced CVaR-RF, which is able to solve CVaR RL for given any reward function after a singular reward-free exploration.
- Proposed CVaR-RF-UCRL as the exploration algorithm and established upper and lower bounds for the sample complexity.
- Developed a CVaR-RF-planning algorithm, equipped with CVaR-VI and CVaR-VI-DISC to generate near-optimal Markov policies solely based on the exploration dataset and given reward function.

Conclusion

- Introduced CVaR-RF, which is able to solve CVaR RL for given any reward function after a singular reward-free exploration.
- Proposed CVaR-RF-UCRL as the exploration algorithm and established upper and lower bounds for the sample complexity.
- Developed a CVaR-RF-planning algorithm, equipped with CVaR-VI and CVaR-VI-DISC to generate near-optimal Markov policies solely based on the exploration dataset and given reward function.
- Demonstrated CVaR-RF-Exploration has the adaptability to different levels of risk tolerance.

Conclusion

- Introduced CVaR-RF, which is able to solve CVaR RL for given any reward function after a singular reward-free exploration.
- Proposed CVaR-RF-UCRL as the exploration algorithm and established upper and lower bounds for the sample complexity.
- Developed a CVaR-RF-planning algorithm, equipped with CVaR-VI and CVaR-VI-DISC to generate near-optimal Markov policies solely based on the exploration dataset and given reward function.
- Demonstrated CVaR-RF-Exploration has the adaptability to different levels of risk tolerance.
- Derived the lower bound for any exploration algorithm in CVaR-RF framework.

Conclusion

- Introduced CVaR-RF, which is able to solve CVaR RL for given any reward function after a singular reward-free exploration.
- Proposed CVaR-RF-UCRL as the exploration algorithm and established upper and lower bounds for the sample complexity.
- Developed a CVaR-RF-planning algorithm, equipped with CVaR-VI and CVaR-VI-DISC to generate near-optimal Markov policies solely based on the exploration dataset and given reward function.
- Demonstrated CVaR-RF-Exploration has the adaptability to different levels of risk tolerance.
- Derived the lower bound for any exploration algorithm in CVaR-RF framework.
- Validated the effectiveness and practicality of our CVaR-RF framework.

Reference

1 Jin, C., Krishnamurthy, A., Simchowitz, M., and Yu, T. Reward-free exploration for reinforcement learning. In International Conference on Machine Learning, pp. 4870– 4879. PMLR, 2020.

2 Baüerle, N. and Ott, J. Markov decision processes with average-value-at-risk criteria. Mathematical Methods of Operations Research, 74:361–379, 2011.

3 Wang, K., Kallus, N., and Sun, W. Near-minimax-optimal risk-sensitive reinforcement learning with CVaR. arXiv preprint arXiv:2302.03201, 2023.