Safe Reinforcement Learning (Constrained MDPs)

Constrained RL

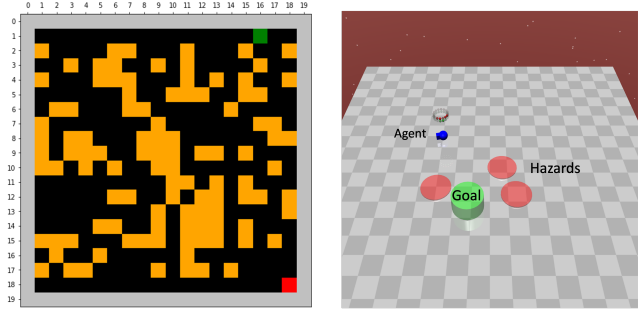


Figure: Grid World and DynamicEnv[1] with Safety Constraints

Constrained MDPs (CMDPs)

Notation

- $\mathcal{S}$ is the state space with $|\mathcal{S}| = S$,
- $\mathcal{A}$ is the action space with $|\mathcal{A}| = A$,
- H is the number of steps in each episode,
- $\mathbb{P} = \{\mathbb{P}_h\}_{h=1}^{H}$ is a collection of transition kernels (transition probability matrices).

Definition: Episodic CMDPs

- At the beginning of each episode, an initial state x_1 is sampled from distribution μ_0.
- At step h, the agent takes action a_h after observing state x_h.
- The agent receives a reward $r_h(x_h, a_h)$ and incurs a utility $g_h(x_h, a_h)$.
- The environment then moves to a new state x_{h+1} sampled from distribution $\mathbb{P}_h(\cdot|x_h, a_h)$. For simplicity, we assume $r_h(x, a)(g_h(x, a)) : \mathcal{S} \times \mathcal{A} \to [0, 1]$.

Policy

A collection of H functions $\{\pi_h : \mathcal{S} \to \mathcal{A}\}_{h=1}^{H}$,

Value Functions and Q-Functions

$$V_h^\pi(x) = \mathbb{E}\left[\sum_{i=h}^{H} r_i(x_i, \pi_i(x_i)) \,\middle|\, x_h = x\right]$$

$$Q_h^\pi(x,a) = r_h(x,a) + \mathbb{E}\left[\sum_{i=h+1}^{H} r_i(x_i, \pi_i(x_i)) \,\middle|\, \begin{array}{c} x_h = x \\ a_h = a \end{array}\right]$$

$$W_h^\pi(x) = \mathbb{E}\left[\sum_{i=h}^{H} g_i(x_i, \pi_i(x_i)) \,\middle|\, x_h = x\right]$$

$$C_h^\pi(x,a) = g_h(x,a) + \mathbb{E}\left[\sum_{i=h+1}^{H} g_i(x_i, \pi_i(x_i)) \,\middle|\, \begin{array}{c} x_h = x \\ a_h = a \end{array}\right].$$

CMDPs

The objective of the agent is to find a policy that maximizes the expected cumulative reward subject to a constraint on the expected utility:

$$\max_{\pi \in \Pi} \mathbb{E}\left[V_1^{\pi}(x_1)\right] \quad \text{subject to: } \mathbb{E}\left[W_1^{\pi}(x_1)\right] \geq \rho, \tag{1}$$

where we assume $\rho \in [0, H]$ to avoid triviality and the expectation is taken with respect to the initial distribution $x_1 \sim \mu_0$.

Solutions with a Given Model

Linear Programming

A CMDP is equivalent to the following linear programming problem.

$$\max_{q_h} \sum_{h,x,a} q_h(x,a) r_h(x,a)$$

$$\text{s.t.:} \sum_{h,x,a} q_h(x,a) g_h(x,a) \geq \rho$$

$$\sum_{a} q_h(x,a) = \sum_{x',a'} p_{h-1}(x|x',a') q_{h-1}(x',a'),$$

$$\sum_{a} q_1(x,a) = \mu_0(x), q_h(x,a) \geq 0$$

$$\sum_{x,a} q_h(x,a) = 1, \forall h \in [H], \forall x \in \mathcal{S}.$$

Solutions with a Given Model

$q_h(x, a)$ that satisfies

$$\sum_a q_h(x, a) = \sum_{x', a'} p_{h-1}(x|x', a') q_{h-1}(x', a'),$$
$$\sum_a q_1(x, a) = \mu_0(x), q_h(x, a) \geq 0$$
$$\sum_{x,a} q_h(x, a) = 1, \forall h \in [H], \forall x \in \mathcal{S}.$$

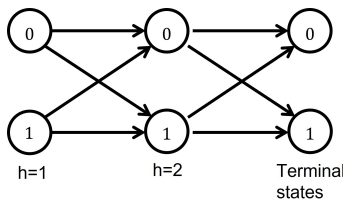
is called an occupancy measure, which is the probability that the agent visits state, action pair (x, a) at step h, i.e. the probability that the agent is in state x at step h and takes action a.

Example

Consider the following simple model with $H = 2$. Assume $\mu_0(x) = (0.5, 0.5)$.

(x,a)	(0,0)	(0, 1)	(1,0)	(1,1)
$P_1(x'\|(x,a))$	(0.8, 0.2)	(0.4, 0.6)	(0.6, 0.4)	(0.1, 0.9)
$P_2(x'\|(x,a))$	(0.8, 0.2)	(0.5, 0.5)	(0.5, 0.5)	(0.1, 0.9)

Table: Transition probabilities at step 1 and step 2



Example

Consider a deterministic policy $\pi_h(a|x)$ such that $\pi_h(a|x) = 1$ for $a = x$. The corresponding occupancy measure is

(x,a)	(0,0)	(0, 1)	(1,0)	(1,1)
$q_1(x, a)$	0.5	0	0	0.5
$q_2(x, a)$	0.45	0	?	?

Table: Occupancy measure

Each policy leads to a unique occupancy measure.

Example

Now given occupancy measure as follows

(x,a)	(0,0)	(0, 1)	(1,0)	(1,1)
$q_1(x, a)$	0.3	0.2	0.1	0.4
$q_2(x, a)$	0.22	0.22	0.29	0.29

Table: Occupancy measure

Example

Now given occupancy measure as follows

(x,a)	(0,0)	(0, 1)	(1,0)	(1,1)
$q_1(x, a)$	0.3	0.2	0.1	0.4
$q_2(x, a)$	0.28	0.14	0.29	0.29

Table: Occupancy measure

Each occupancy measure leads to a unique policy:

$$\pi_h(a|x) = \frac{q_h(x, a)}{\sum_u q_h(x, u)}.$$

Each occupancy measure leads to a unique policy:

$$\pi_h(a|x) = \frac{q_h(x,a)}{\sum_u q_h(x,u)}.$$

$(a\|x)$	$(0\|0)$	$(1\|0)$	$(0\|1)$	$(1\|1)$
$\pi_1(a\|x)$	0.6	0.4	0.2	0.8
$\pi_2(a\|x)$	0.67	0.33	0.5	0.5

Table: Policy induced from the occupancy probability

Each policy leads to a unique occupancy measure.

Each occupancy measure leads to a unique policy.

Therefore, finding the optimal policy is equivalent to solving the linear programming.

How about value iteration or policy iteration like for unconstrained MDPs?

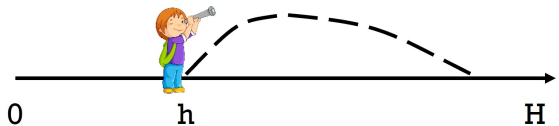
The Fundamental Differences: Principle of Optimality

Unconstrained

Principle of Optimality: A subsolution of an optimal solution is the optimal solution to the corresponding subproblem.

The Bellman equation:

$$Q_h(x_h, a_h) = r_h(x_h, a_h) + E\left[\max_v Q_{h+1}(x_{h+1}, v))\right].$$



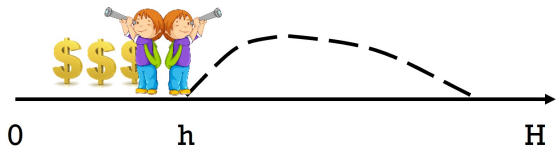
The Fundamental Differences: Principle of Optimality

Constrained

~~Principle of Optimality:~~ A subsolution of an optimal solution is the optimal solution to the corresponding subproblem.

~~The Bellman equation:~~

$$Q_h(x_h, a_h) = r_h(x_h, a_h) + E\left[\max_v Q_{h+1}(x_{h+1}, v))\right].$$



The Fundamental Differences

- Difference 1: The optimal policy is stochastic.
- The regret is $O(\sqrt{T})$ ($O(\log T)$ regret is unachievable).

Model-based Solution

Notation

$q_h(x, a; \mathbf{p})$: An occupancy probability that is compatible with transition kernel $\mathbf{p}$.

$$\sum_a q_h(x, a) = \sum_{x', a'} p_{h-1}(x|x', a') q_{h-1}(x', a'),$$

$$\sum_a q_1(x, a) = \mu_0(x), q_h(x, a) \geq 0$$

$$\sum_{x,a} q_h(x, a) = 1, \forall h \in [H], \forall x \in \mathcal{S}.$$

Model-based Solution

OptCMDP-Bonus (Yonathan, Mannor, Pirotta 2020)

Estimate the transition kernel, reward function and utility function by empirical means:

$$\bar{p}_{k,h}(x'|(x,a)) = \frac{\sum_{l=1}^{k} \mathbb{I}(x_{l,h} = x, a_{l,h} = a, x_{l,h+1} = x')}{\sum_{l=1}^{k} \mathbb{I}(x_{l,h} = x, a_{l,h} = a)}$$

$$\bar{r}_{k,h}(x,a) = \frac{\sum_{l=1}^{k} r_{l,h} \mathbb{I}(x_{l,h} = x, a_{l,h} = a)}{\sum_{l=1}^{k} \mathbb{I}(x_{l,h} = x, a_{l,h} = a)}$$

$$\bar{r}_{k,h}(x,a) = \frac{\sum_{l=1}^{k} g_{l,h} \mathbb{I}(x_{l,h} = x, a_{l,h} = a)}{\sum_{l=1}^{k} \mathbb{I}(x_{l,h} = x, a_{l,h} = a)}$$

Define the exploration bonus to be:

$$b_{k,h}(x,a) = c\sqrt{\frac{1}{N_{k,h}(x,a)}} + c' H \sum_{x'} \left(\sqrt{\frac{\bar{p}_{k,h}(x'|(x,a))(1 - \bar{p}_{k,h}(x'|(x,a)))}{N_{k,h}(x,a)}} + \frac{1}{N_{k,h}(x,a)} \right).$$

Model-based Solution

OptCMDP-Bonus (Yonathan, Mannor, Pirotta 2020)

At the beginning of episode k, solve the following linear programming problem:

$$\max_{q} \sum_{h,x,a} q_h(x,a;\bar{\mathbf{p}}) \left(\bar{r}_{k-1,h}(x,a) + b_{k-1,h}(x,a)\right)$$

$$\text{s.t.:} \sum_{h,x,a} q_h(x,a;\bar{\mathbf{p}}) \left(\bar{g}_{k-1,h}(x,a) + b_{k-1,h}(x,a)\right) \geq \rho$$

Claim

π_k (the solution to the optimization problem above) is an optimistic policy, i.e.

$$V_1^{\pi_k}(x_1; \bar{\mathbf{r}}_{k-1} + \mathbf{b}_{k-1}, \bar{\mathbf{p}}_{k-1}) \geq V_1^{\pi^*}(x_1).$$

Regret Analysis

OptCMDP-Bonus (Yonathan, Mannor, Pirotta 2020)

Reward regret:

$$\text{Regret}_K = \sum_{k=1}^{K} \left(V_1^{\pi^*}(x_1) - V_1^{\pi_k}(x_1) \right)^+ = \tilde{\mathcal{O}}(\sqrt{K}).$$

Constraint violation:

$$\text{Violation}_K = \sum_{k=1}^{K} (\rho - W_1^{\pi_k}(x_1))^+ = \tilde{\mathcal{O}}(\sqrt{K}).$$

Model-based Primal-Dual Solution for CMDPs

Linear Programming for CMDPs

$$\max_{q} \sum_{h,x,a} q_h(x,a;\mathbf{p}) r_h(x,a)$$

$$\text{s.t.:} \sum_{h,x,a} q_h(x,a;\mathbf{p}) g_h(x,a) \geq \rho$$

The dual problem

$$\min_{\lambda \geq 0} \max_{q} \sum_{h,x,a} q_h(x,a;\mathbf{p}) r_h(x,a) + \lambda \left(\sum_{h,x,a} q_h(x,a;\mathbf{p}) g_h(x,a) - \rho \right)$$

Model-based Primal-Dual Solution for CMDPs

OptPrimalDual-CMDP

At the beginning of each episode

- Step 1: Compute exploration bonus

$$b_{k-1,h}(x,a) = c\sqrt{\frac{1}{N_{k,h}(x,a)}} + c' H \sum_{x'} \left(\sqrt{\frac{Var(\bar{p}_{k,h}(x'|(x,a)))}{N_{k,h}(x,a)}} + \frac{1}{N_{k,h}(x,a)} \right),$$

$$\tilde{r}_{k-1,h}(x,a) = \bar{r}_{k-1,h}(x,a) + b_{k-1,h}(x,a)$$
$$\tilde{g}_{k-1,h}(x,a) = \bar{g}_{k-1,h}(x,a) + b_{k-1,h}(x,a),$$

and the dual variable

$$\lambda_{k-1} = \lambda_{k-2} + \frac{1}{\eta} \left(\rho - \sum_{h,x,a} q_{k-1,h}(x,a;\bar{p}_{k-1},\pi_{k-1}) \tilde{g}_{k-1,h}(x,a) \right) \Bigg|_0^{\rho}$$

Model-based Primal-Dual Solution for CMDPs

OptPrimalDual-CMDP

- Step 2: Update the reward Q-values $\tilde{Q}_h^{\pi_{k-1}}(x,a)$ and utility Q-values $\tilde{C}_h^{\pi_{k-1}}(x,a)$ based on $\tilde{r}_{k-1}$, $\tilde{g}_{k-1}$, $\bar{p}_{k-1}$ and π_{k-1} using the backward computation of dynamic programming.
- Step 3: Compute the pseudo Q function

$$Q_{k,h}(x,a) = \tilde{Q}_h^{\pi_{k-1}}(x,a) + \lambda_{k-1}\tilde{C}_h^{\pi_{k-1}}(x,a)$$

  and policy

$$\pi_{k,h}(a|x) = \frac{\pi_{k-1,h}(a|x)\exp\left(-\beta Q_{k,h}(x,a)\right)}{\sum_u \pi_{k-1,h}(u|x)\exp\left(-\beta Q_{k,h}(x,u)\right)}.$$

- Step 4: Execute policy π_k to collect $\{x_{k,h}, a_{k,h}, r_{k,h}, g_{k,h}\}$ for $h = 1, \cdots, H$.

$\eta = \sqrt{\frac{H^2K}{\rho^2}}$ and $\beta = \sqrt{\frac{2\log A}{H^2(1+\rho)^2K}}$

Regret Analysis

OptPrimalDual-CMDP (Yonathan, Mannor, and Pirotta, 2020)

Reward regret:

$$\text{Regret}_K = \tilde{\mathcal{O}}(\sqrt{K}).$$

Constraint violation:

$$\text{Violation}_K = \tilde{\mathcal{O}}(\sqrt{K}).$$

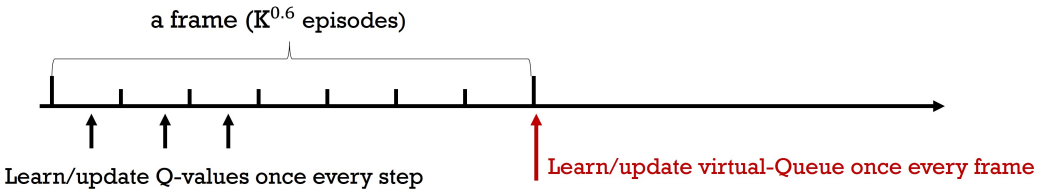
Model-Free Primal-Dual Solution for CMDPs

Triple-Q

Triple-Q maintains three "Q" values:

- $Q_{k,h}(x, a)$: Q-value for the reward
- $C_{k,h}(x, a)$: Q-value for the utility
- λ : virtual-Queue (dual variable)

Two-time-scale implementation: Consider K episodes. Group every $K^{0.6}$ episodes as a frame.



Model-Free Primal-Dual Solution for CMDPs

Triple-Q

At the beginning of each frame

- Update virtual-queue:

$$\lambda_F = \left(\lambda_{F-1} + \rho + \epsilon - \frac{\sum_{k \in \text{frame } F-1} C_{k,1}(x_{k,1}, a_{k,1})}{K^{0.6}} \right)_0$$

- Reset the reward Q-values

$$C_{k,h}(x, a) = Q_{k,h}(x, a) = H,$$

and all visit counts to zero.

Model-Free Primal-Dual Solution for CMDPs

Triple-Q

At each step of episode k in frame F, use UCB-SARSA, i.e.

- Select an action based on the pseudo-Q value

$$a_{k,h} \in \max_u \left(Q_{k,h}(x_{k,h}, u) + \frac{\lambda_F}{\eta} C_{k,h}(x_{k,h}, u) \right)$$

- With data $(x_{h-1} = x, u_{h-1} = u, x_h, u_h)$ and $r_h(x_{h-1}, u_{h-1}) = r$, update reward and utility Q-functions

$$\begin{aligned} Q_{k+1,h-1}(x, u) &= Q_{k,h-1}(x, u) + \beta_{k,h}(r + Q_{k,h}(x_h, u_h) + b_{k,h-1} - Q_{k,h-1}(x, u)) \\ C_{k+1,h-1}(x, u) &= C_{k,h-1}(x, u) + \beta_{k,h}(r + C_{k,h}(x_h, u_h) + b_{k,h-1} - C_{k,h-1}(x, u)) \end{aligned}$$

$b_{k,h}$ is the simple UCB bonus similar to the one in UCB-SARAR.

Regret Analysis

Triple-Q (Wei, Liu, and Ying, 2022)

Reward regret:

$$\text{Regret}_K = \tilde{\mathcal{O}}(K^{0.8}).$$

Constraint violation:

$$\text{Violation}_K = 0.$$

Reference

- Efroni Yonathan, Shie Mannor, and Matteo Pirotta. "Exploration-exploitation in constrained MDPs." arXiv preprint arXiv:2003.02189 (2020).
- H. Wei, X. Liu, and L. Ying. "Triple-Q: A Model-Free Algorithm for Constrained Reinforcement Learning with Sublinear Regret and Zero Constraint Violation", AISTATS 2022.