Safe Reinforcement Learning with Linear Function Approximation

Sanae Amani [1] Christos Thrampoulidis [2] Lin F. Yang[1]

[1] University of California, Los Angeles

[2]University of British Columbia, Vancouver

July 2021

Table of Contents

Problem Formulation

- **Finite horizon MDP:** $M = (\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r, c)$, $\mathcal{S}$: known state set, $\mathcal{A}$: known action set, H: known episode's length, $\mathbb{P} = \{\mathbb{P}_h\}_{h=1}^{H}$: unknown transition probabilities, $r = \{r_h\}_{h=1}^{H}$: unknown reward functions, and $c = \{c_h\}_{h=1}^{H}$: unknown cost functions.

Problem Formulation

- **Finite horizon MDP:** $M = (\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r, c)$, $\mathcal{S}$: known state set, $\mathcal{A}$: known action set, H: known episode's length, $\mathbb{P} = \{\mathbb{P}_h\}_{h=1}^H$: unknown transition probabilities, $r = \{r_h\}_{h=1}^H$: unknown reward functions, and $c = \{c_h\}_{h=1}^H$: unknown cost functions.
- **Safety Constraint:** When being in state s_h^k, at episode k and time-step $h \in [H]$, the agent must select a *safe* policy π_h^k such that
  - if π_h^k is deterministic:

$$c_h(s_h^k, \pi_h^k(s_h^k)) \leq \tau.$$

Problem Formulation

- **Finite horizon MDP:** $M = (\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r, c)$, $\mathcal{S}$: known state set, $\mathcal{A}$: known action set, H: known episode's length, $\mathbb{P} = \{\mathbb{P}_h\}_{h=1}^H$: unknown transition probabilities, $r = \{r_h\}_{h=1}^H$: unknown reward functions, and $c = \{c_h\}_{h=1}^H$: unknown cost functions.
- **Safety Constraint:** When being in state s_h^k, at episode k and time-step $h \in [H]$, the agent must select a *safe* policy π_h^k such that
  - if π_h^k is deterministic:

$$c_h(s_h^k, \pi_h^k(s_h^k)) \leq \tau.$$

  - if π_h^k is randomized:

$$\mathbb{E}_{a \sim \pi_h^k(s_h^k)} c_h(s_h^k, a) \leq \tau.$$

Problem Formulation

Goal

$$V_h^*(s) = \sup_{\pi \in \Pi^{\mathrm{safe}}} V_h^\pi(s), \ \forall (s,h) \in \mathcal{S} \times [H]$$

$$R_K := \sum_{k=1}^{K} V_1^*(s_1^k) - V_1^{\pi^k}(s_1^k).$$

The agent's goal is to keep R_K as small as possible, while π^k are safe for all $k \in [K]$ with high probability.

Key Assumptions

- M is a linear MDP with feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, if for any $h \in [H]$, there exist d unknown measures $\boldsymbol{\mu}_h^* := [\mu_h^{*(1)}, \ldots, \mu_h^{*(d)}]^\top$ over $\mathcal{S}$, and unknown vectors $\boldsymbol{\theta}_h^*, \boldsymbol{\gamma}_h^* \in \mathbb{R}^d$ such that $\mathbb{P}_h(.|s,a) = \langle \boldsymbol{\mu}_h^*(.), \phi(s,a) \rangle$, $r_h(s,a) = \langle \boldsymbol{\theta}_h^*, \phi(s,a) \rangle$, and $c_h(s,a) = \langle \gamma_h^*, \phi(s,a) \rangle$.

Key Assumptions

- M is a linear MDP with feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, if for any $h \in [H]$, there exist d unknown measures $\boldsymbol{\mu}_h^* := [\mu_h^{*(1)}, \ldots, \mu_h^{*(d)}]^\top$ over $\mathcal{S}$, and unknown vectors $\boldsymbol{\theta}_h^*, \boldsymbol{\gamma}_h^* \in \mathbb{R}^d$ such that $\mathbb{P}_h(.|s, a) = \langle \boldsymbol{\mu}_h^*(.), \phi(s, a) \rangle$, $r_h(s, a) = \langle \boldsymbol{\theta}_h^*, \phi(s, a) \rangle$, and $c_h(s, a) = \langle \boldsymbol{\gamma}_h^*, \phi(s, a) \rangle$.
- For all $s \in \mathcal{S}$, there exists a known safe action $a_0(s)$ with known safety measure $\tau_h(s) := \langle \phi\left(s, a_0\left(s\right)\right), \boldsymbol{\gamma}_h^* \rangle < \tau$ for all $h \in [H]$.

Table of Contents

SLUCB-QVI and RSLUCB-QVI

- Algorithms for deterministic and randomized policy selection.

SLUCB-QVI and RSLUCB-QVI

- Algorithms for deterministic and randomized policy selection.
- They run LSVI to compute estimated Q functions and inner approximation safe policy set.

SLUCB-QVI and RSLUCB-QVI

- Algorithms for deterministic and randomized policy selection.
- They run LSVI to compute estimated Q functions and inner approximation safe policy set.
- They achieve a $\tilde{\mathcal{O}}\left(\kappa\sqrt{d^3H^3T}\right)$ regret, nearly matching that of state-of-the-art unsafe algorithms, where

$$\kappa := \arg\max_{h,s} \frac{2H}{\tau - \tau_h(s)} + 1$$

is a constant characterizing the safety constraints.