Near-Optimal Algorithms for Autonomous Exploration and Multi-Goal Stochastic Shortest Path

Haoyuan Cai, Tsinghua University

Tengyu Ma, Stanford University

Simon Du, University of Washington

Introduction

- We study the incremental autonomous exploration problem.
- Large state space, unknown environment
- Expand the range of known states, learn near-optimal policies
- Applications: navigation in mazes, game playing and so on.

Problem Definition

- MDP $M = \langle \mathcal{S}, \mathcal{A}, P, c, s_0 \rangle$, Policy $\pi : \mathcal{S} \to \mathcal{A}$
- For any $g \in \mathcal{S}$, denote $t_g^\pi(s) := \inf \{ t \geq 0 : s_{t+1} = g \mid s_1 = s, \pi \}$.
- Denote $V_g^\pi(s)$ as the expected cost to reach g from s using policy π.

$$V_g^\pi(s) = \mathbb{E}\left[\sum_{t=1}^{t_g^\pi(s)} c_t\left(s_t, \pi\left(s_t\right)\right) \mid s_1 = s \right],$$

$$Q_g^\pi(s, a) = \mathbb{E}\left[\sum_{t=1}^{t_g^\pi(s)} c_t\left(s_t, \pi\left(s_t\right)\right) \mid s_1 = s, \pi(s_1) = a \right].$$

- Exploration radius: L
- Objective: learn the set of incrementally controllable states $\mathcal{S}_L^{\to}$

Problem Definition: Multi-Goal Stochastic Shortest Path

- MDP $M = \langle \mathcal{S}, \mathcal{A}, P, c, s_0 \rangle$, and $\mathcal{S}_L^{\rightarrow} = \mathcal{S}$
- Input: error ε, confidence $\delta \in (0,1)$, goal space $\mathcal{G} \subseteq \mathcal{S}$
- Output: a set of policies $\{\pi_s\}_{s \in \mathcal{G}}$, s.t.

$$\forall s \in \mathcal{G}, V_s^{\pi_s}(s_0) \leq V_s^*(s_0) + \varepsilon L.$$

- Denote T as the total number of steps the agent uses
- Use $C_T := \sum_{t=1}^{T} c_t(s_t, a_t)$ to measure the performance

Problem Definition: Autonomous Exploration (AX)

- MDP $M = \langle \mathcal{S}, \mathcal{A}, P, c, s_0 \rangle$
- Input: exploration radius L, error ε, confidence $\delta \in (0, 1)$
- Output: a set of states $\mathcal{K} \supseteq \mathcal{S}_L^{\rightarrow}$ and a set of policies $\{\pi_s\}_{s \in \mathcal{K}}$, s.t.

$$\forall s \in \mathcal{S}_L^{\rightarrow}, V_s^{\pi_s}(s_0) \leq (1 + \varepsilon)L.$$

Comparison

Algorithm	Sample Complexity
UcbExplore (Lim & Auer, 2012)	$\widetilde{O}(L^3 S^2 A/\varepsilon^3)$
DisCo (Tarbouriech et al., 2020)	$\widetilde{O}\left(L^3 S^2 A/\varepsilon^2\right)$
VALAE	$\widetilde{O}\left(LSA/\varepsilon^2\right)$
Lower Bound	$\Omega(LSA/\varepsilon^2)$

Table: Comparisons between our results and prior results.

1: **Input:** Confidence $\delta \in (0,1)$, error $\varepsilon \in (0,1]$, and $L \geq 1$.
2: **Input (for multi-goal SSP only):** Goal Space $\mathcal{G} \subseteq \mathcal{S}$.
3: (For autonomous exploration, set $\mathcal{G} = \emptyset$.)
4: **Specify:** Trigger set $\mathcal{N} \leftarrow \{2^{j-1} : j = 1, 2, \ldots\}$.
\\We run DisCo algorithm with $\varepsilon = 1$ and get a set $\mathcal{K}$ such that $\mathcal{S}_L^{\rightarrow} \subseteq \mathcal{K} \subseteq \mathcal{S}_{2L}^{\rightarrow}$.
5: Run DisCo algorithm with input $(\delta, \varepsilon = 1, L)$ and we get a set $\mathcal{K}$ and a set of policies $\{\pi_s\}_{s \in \mathcal{K}}$.
6: Run Alg. 1 with input $(\delta, L, \mathcal{K}, \{\pi_s\}_{s \in \mathcal{K}})$, and we obtain the variables $N(), n(), \widehat{P}, \theta(), \widehat{c}$.
7: Set time step $t \leftarrow 1$ and trigger index $j \leftarrow 5 + \log_2 \frac{1}{c_{\min}}$.
8: Set $\epsilon \leftarrow \varepsilon/3$, $B \leftarrow 10L$, $\lambda = \widetilde{O}(1/\epsilon^2)$, and $g \leftarrow s_0$.
9: Initialize $\mathcal{G} \leftarrow \mathcal{K}$ if $\mathcal{G} = \emptyset$.
10: *\\Solve multi-goal SSP problem on $M^\dagger$ with goal space $\mathcal{G}$.*

- Step 1: run DisCo with $\varepsilon = 1$
  discover a set $\mathcal{K}$ s.t. $\mathcal{S}_L^{\rightarrow} \subseteq \mathcal{K} \subseteq \mathcal{S}_{2L}^{\rightarrow}$
- Step 2: reduce AX to multi-goal SSP
  merge all $s \notin \mathcal{K}$, construct MDP $M^\dagger$, collect samples for all $(s, a) \in \mathcal{K} \times \mathcal{A}$

Our Algorithm

11: **for** round $r = 1, 2, \cdots$ **do**
12: *Phase (a): Compute Optimal Policy*
13: Compute $(Q, V) := \texttt{VISGO}(g, 2^{-j}/(|\mathcal{K}^\dagger|A))$.
14: Set the policy $\tilde{\pi}$ as the greedy policy over Q, and $\hat{\tau} \leftarrow 0$.
15: *Phase (b): Policy Evaluation*
16: **for** episode $k = 1, 2, \cdots, \lambda$ **do**
17: Set $s_t \leftarrow s_0$ and reset to the initial state s_0, and $\hat{\tau}_k \to 0$.
18: **while** $s_t \neq g$ **do**
19: Take action $a_t = \arg\min_{a \in \mathcal{A}} Q(s_t, a)$ on $M^\dagger$, incur cost c_t and observe next state $s_{t+1} \sim P^\dagger(\cdot \mid s_t, a_t)$.
20: Set $(s, a, s', c) \leftarrow (s_t, a_t, s_{t+1}, c_t)$ and $t \leftarrow t + 1$.
21: Set $N(s, a) \leftarrow N(s, a) + 1$, $\theta(s, a) \leftarrow \theta(s, a) + c$, $N(s, a, s') \leftarrow N(s, a, s') + 1$.
22: **if** $N(s, a) \in \mathcal{N}$ **then**
23: Set $j \leftarrow j + 1$, $\widehat{c}(s, a) \leftarrow \frac{2\theta(s,a)}{N(s,a)}$ and $\theta(s, a) \leftarrow 0$.
24: For all $s' \in \mathcal{K}^\dagger$, set $n(s, a) \leftarrow N(s, a)$, $\widehat{P}_{s,a,s'} \leftarrow N(s, a, s')/N(s, a)$.
25: Return to line 11, start a new round (the current round has been a *skipped round*).
26: **end if**
27: Set $\hat{\tau} \leftarrow \hat{\tau} + \frac{c}{\lambda}$, $\hat{\tau}_k \leftarrow \hat{\tau}_k + c$.
28: **end while**
29: **if** $\hat{\tau} > V(s_0) + \epsilon L$ **then**
30: Return to line 11, start a new round. (the current round has been a *failure round*).
31: **end if**
32: **end for**
33: Set $\pi_g \leftarrow \tilde{\pi}$. Remove g from $\mathcal{G}$. (The current round has been a *success round*.)
34: Choose another state $g \in \mathcal{G}$.
35: Stop the algorithm if $\mathcal{G}$ is empty.
36: **end for**

- Step 3: solve multi-goal SSP on $M^\dagger$

Phase (a):
compute optimal policy $\tilde{\pi}$ with goal g

Phase (b):
execute $\tilde{\pi}$ for $\lambda = \widetilde{O}(1/\epsilon^2)$ times

Our New Techniques

- **Connection between AX and Multi-Goal SSP**
  Intuition: exploit variance information in value functions
  extend Bernstein-type bounds from single-goal SSP to multi-goal SSP

- **Using Regret to Bound the Sample Complexity**
  Intuition: use regret to bound the total number of rounds r

  For the upper bound, we extend variance analysis from classical SSP. For the lower bound, the total regret in all the failure rounds grows linearly, and we use concentration inequalities to lower bound the total regret in success rounds and skipped rounds.

Lower Bound

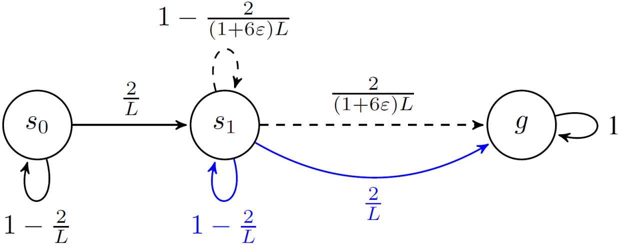

Figure: Illustration of our construction of the hard MDP.

Action a^* in s_1: the blue edges   Other actions in s_1: the dashed edges

If $\pi_g(s_1) = a^*$, we have $V_g^{\pi_g}(s_0) = L$. Otherwise, $V_g^{\pi_g}(s_0) > (1+\varepsilon)L$.
To discriminate two Bernoulli distributions with $p_1 = \frac{2}{L}$ and $p_2 = \frac{2}{(1+6\varepsilon)L}$ among all the A actions, the algorithm needs $\widetilde{\Omega}(LA/\varepsilon^2)$ samples.

Summary

- New Algorithm: `Value-Aware Autonomous Exploration` (`VALAE`)
  - First algorithm enjoying near-optimal sample complexity bound $\widetilde{O}\left(LSA/\varepsilon^2\right)$
  - Use DisCo as initial steps and use the estimated value functions to guide our exploration
  - Connect autonomous exploration to multi-goal stochastic shortest path
  - New analysis techniques: using concentration inequalities to lower bound the regret
- First lower bound for autonomous exploration: $\Omega(LSA/\varepsilon^2)$
  - Use the techniques of KL divergence

Thanks for listening!