人工智慧作業報告簡報

信息工程學院 2021 級 干皓丞 2101212850

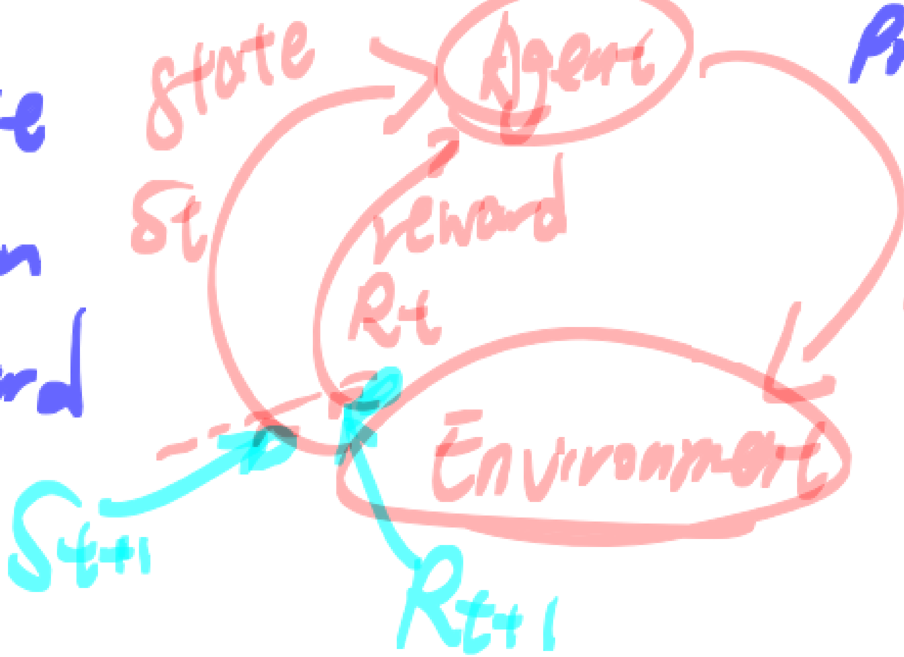
Planning in Markov Decision Processes with Gap-Dependent Sample Complexity
用間隙相關樣本複雜性來做出利用馬爾可夫決策過程所做出的規劃
https://arxiv.org/abs/2006.05879
https://proceedings.neurips.cc/paper/2020
https://proceedings.neurips.cc/paper/2020/hash/0d85eb24e2add96ff1a7021f83c
1abc9-Abstract.html
Part of Advances in Neural Information Processing Systems 33 (NeurIPS 2020)

Motivation 动机

強化學習 3要素
(1) State
(2) Action
(3) Reward

2.特性
(1) 所有過程隨機性
(2) 下一個狀態轉移只與現在的狀態有關

Markov Decision Process



MDP.

狀態 State, $S=\{s_0, s_1, \ldots, s_n\}$
動作 Action, $A(s)=\{a_0, a_1, \ldots, a_n\}$ → $M=(S, A, P, r)$.
獎勵 Reward, $R=\{r_0, r_1, \ldots, r_n\}$.
轉移概率 Dynamic, $P=p(s', r \mid s, a)$, for all $s \in S$, $a \in A$

該研究的研究者提出名為一個名為 MDP-GapE 的算法，這個演算法是一種新的基於“軌跡的蒙特卡洛樹搜索演算法“(trajectory-based Monte-Carlo Tree Search algorithm)，此算法用於在馬爾可夫決策過程中進行規劃，該算法會在過程中進行規畫，且而這個過程的轉換是具有‘有限支持‘(a finite support)。

Intuition 直觉

Table 1: Different settings of planning algorithms in the literature

Setting	Input	Output	Optimality criterion
(1) Fixed confidence (action-based)	ε, δ	$\widehat{a}_n$	$\mathbb{P}\left(\bar{r}_n(\widehat{a}_n) \leq \varepsilon\right) \geq 1-\delta$
(2) Fixed confidence (value-based)	ε, δ	$\widehat{V}(s_1)$	$\mathbb{P}\left(\lvert\widehat{V}(s_1) - V^\star(s_1)\rvert \leq \varepsilon\right) \geq 1-\delta$
(3) Fixed budget	n (budget)	$\widehat{a}_n$	$\mathbb{E}\left[\bar{r}_n(\widehat{a}_n)\right]$ decreasing in n
(4) Anytime	-	$\widehat{a}_n$	$\mathbb{E}\left[\bar{r}_n(\widehat{a}_n)\right]$ decreasing in n

We propose an algorithm in the *fixed confidence* setting (ε, δ): after n calls to the generative model, the algorithm should return an action $\hat{a}_n$ such that $\bar{r}_n(\hat{a}_n) \leq \varepsilon$ with probability at least $1-\delta$. We prove that its *sample complexity* n is bounded in high probability by a quantity that depends on

the sub-optimality gaps of the actions that are applicable in state s_1. We also provide experiments showing its effectiveness. The only assumption that we make on the MDP is that the support of the transition probabilities $p_h(\cdot|s, a)$ should have cardinality bounded by $B < \infty$, for all s, a and h.

Justification 理由

Table 2: Algorithms with sample complexity guarantees

Algorithm	Setting	Sample complexity	Remarks
Sparse Sampling [19]	(1)-(2)	$H^5(BK)^H/\varepsilon^2$ or $\varepsilon^{-\left(2+\frac{\log(K)}{\log(1/\gamma)}\right)}$	proved in Lemma 1
OLOP [2]	(3)	$\varepsilon^{-\max\left(2,\frac{\log\kappa}{\log(1/\gamma)}\right)}$	open loop, $\kappa \in [1, K]$
OP [3]	(4)	$\varepsilon^{-\frac{\log\kappa}{\log(1/\gamma)}}$	known MDP, $\kappa \in [0, BK]$
BRUE [8]	(4)	$H^4(BK)^H/\Delta^2$	minimal gap Δ
StOP [27]	(1)	$\varepsilon^{-\left(2+\frac{\log\kappa}{\log(1/\gamma)}+o(1)\right)}$	$\kappa \in [0, BK]$
TrailBlazer [13]	(2)	$\varepsilon^{-\max\left(2,\frac{\log(B\kappa)}{\log(1/\gamma)}+o(1)\right)}$	$\kappa \in [1, K]$
SmoothCruiser [14]	(2)	ε^{-4}	only regularized MDPs
MDP-GapE (ours)	(1)	$\sum_{a_1\in\mathcal{A}} \frac{H^2(BK)^{H-1}B}{(\Delta_1(s_1,a_1)\vee\Delta\vee\varepsilon)^2}$	see Corollary 1

Corollary 1. *The number of episodes used by MDP-GapE satisfies*

$$\mathbb{P}\left(\tau = \mathcal{O}\left(\sum_{a_1} \frac{(BK)^{H-1}}{(\Delta_1(s_1,a_1)\vee\Delta\vee\varepsilon)^2}\left[\log\left(\frac{1}{\delta}\right) + BH\log(BK)\right]\right)\right) \geq 1-\delta\,.$$

Framework 框架

Algorithm 1 MDP-GapE

Input: confidence level δ, tolerance ε
initialize data lists $\mathcal{D}_h \leftarrow [\,]$ for all $h \in [H]$
for $t = 1 \ldots$ **do**
  //Update confidence bounds
  $U_h^{t-1}, L_h^{t-1} \leftarrow \texttt{UpdateBounds}(t, \delta, \mathcal{D}_h)$
  if $U_1^{t-1}(s_1, c^t) - L_1^{t-1}(s_1, b^t) \leq \varepsilon$ **then**
    return b_{t-1}, **break**
  end if
  // Best
  $b^{t-1} \leftarrow \underset{b}{\operatorname{argmin}} \left[\max_{a \neq b} U_1^{t-1}(s_1, a) - L_1^{t-1}(s_1, b)\right]$
  //Challenger
  $c^{t-1} \leftarrow \underset{c \neq b^t}{\operatorname{argmax}}\, U_1^{t-1}(s_1, c)$
  //Exploration
  $a_1^t \leftarrow \underset{a \in \{b^{t-1}, c^{t-1}\}}{\operatorname{argmax}} \left[U_1^{t-1}(s_1, a) - L_1^{t-1}(s_1, a)\right]$
  observe reward r_1^t, next state s_2^t, save $\mathcal{D}_1$.append$(s_1^t, a_1^t, s_2^t, r_1^t)$
  for step $h = 2, \ldots, H$ **do**
    $a_h^t \leftarrow \underset{a}{\operatorname{argmax}}\, U_h^{t-1}(s_h^t, a)$
    observe reward r_{h-1}^t, next state s_h^t,save $\mathcal{D}_h$.append$(s_h^t, a_h^t, s_{h+1}^t, r_h^t)$
  end for
end for

Result 结果

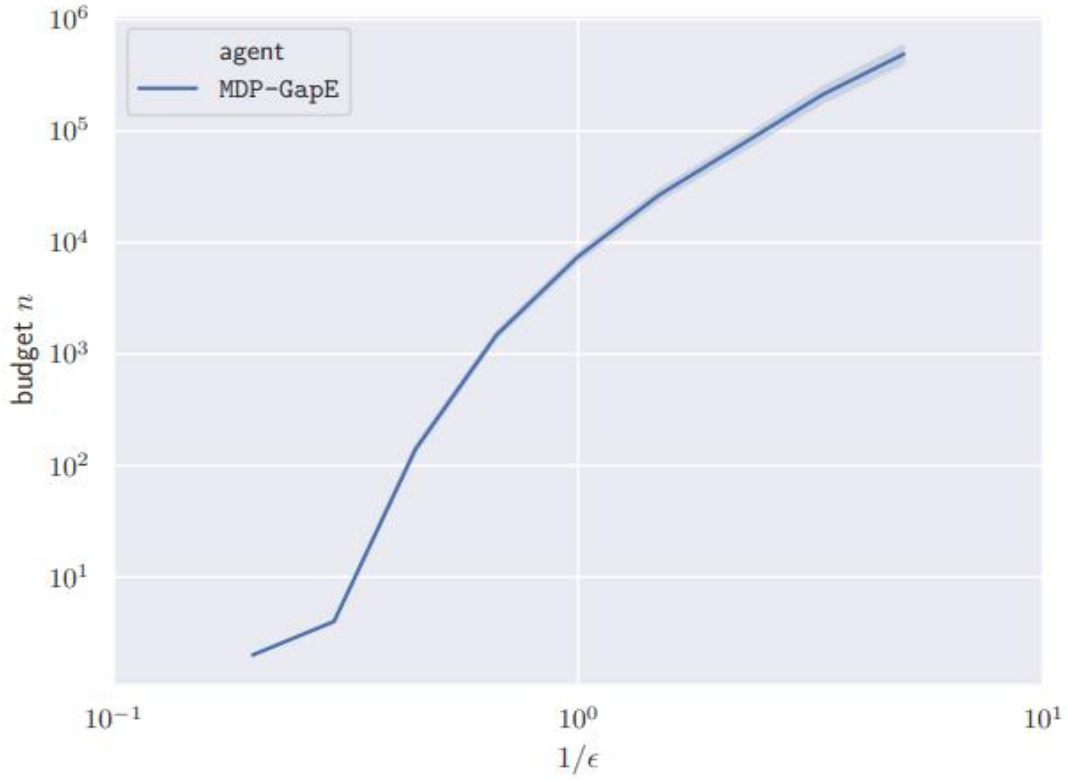


Figure 1: Polynomial dependency of the number n of oracle calls with respect to $1/\varepsilon$.

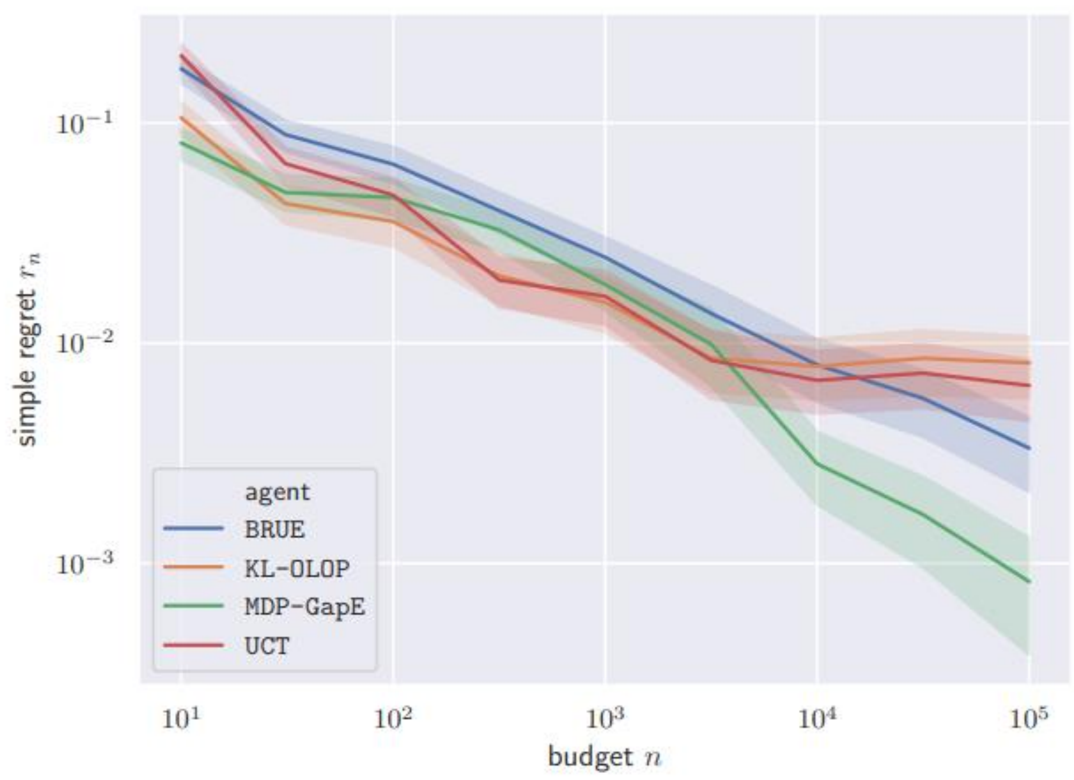


Figure 2: Comparison to KL-OLOP in a fixed-budget setting.