RL Reading Group (IV)

PAC-MDP

Kaiyuan Xu

xky@bu.edu

PAC-MDP (1.5, P2418)

Definition 2 *An algorithm $\mathcal{A}$ is said to be an* **efficient PAC-MDP** *(Probably Approximately Correct in Markov Decision Processes) algorithm if, for any $\varepsilon > 0$ and $0 < \delta < 1$, the per-timestep computational complexity, space complexity, and the sample complexity of $\mathcal{A}$ are less than some polynomial in the relevant quantities $(S, A, 1/\varepsilon, 1/\delta, 1/(1-\gamma))$, with probability at least $1-\delta$. It is simply* **PAC-MDP** *if we relax the definition to have no computational complexity requirement.*

we consider the relaxed but still challenging and useful goal of acting near-optimally on all but a polynomial number of steps

PAC-MDP Analysis Framework

Theorem 10 *Let $\mathcal{A}(\varepsilon,\delta)$ be any greedy learning algorithm such that, for every timestep t, there exists a set K_t of state-action pairs that depends only on the agent's history up to timestep t. We assume that $K_t = K_{t+1}$ unless, during timestep t, an update to some state-action value occurs or the escape event A_K happens. Let M_{K_t} be the known state-action MDP and π_t be the current greedy policy, that is, for all states s, $\pi_t(s) = \text{argmax}_a Q_t(s,a)$. Furthermore, assume $Q_t(s,a) \leq V_{\max}$ for all t and (s,a). Suppose that for any inputs ε and δ, with probability at least $1-\delta$, the following conditions hold for all states s, actions a, and timesteps t: (1) $V_t(s) \geq V^*(s) - \varepsilon$ (optimism), (2) $V_t(s) - V^{\pi_t}_{M_{K_t}}(s) \leq \varepsilon$ (accuracy), and (3) the total number of updates of action-value estimates plus the number of times the escape event from K_t, A_K, can occur is bounded by $\zeta(\varepsilon,\delta)$ (learning complexity). Then, when $\mathcal{A}(\varepsilon,\delta)$ is executed on any MDP M, it will follow a 4ε-optimal policy from its current state on all but*

$$O\left(\frac{V_{\max}\zeta(\varepsilon,\delta)}{\varepsilon(1-\gamma)}\ln\frac{1}{\delta}\ln\frac{1}{\varepsilon(1-\gamma)}\right)$$

timesteps, with probability at least $1-2\delta$.

Sample Complexity Analysis

- Constructing a set of known state-action pairs K_t
- Define an event E that satisfies Condition 2
- Determine parameter m to guarantee event E will occur with probability at least 1-δ
- Determine learning complexity ζ
- Applying Theorem 10

R-MAX

Algorithm 1 R-MAX

Inputs: S, A, γ, m, ε_1, and $U(\cdot,\cdot)$
for all (s,a) **do**
  $Q(s,a) \leftarrow U(s,a)$ // *action-value estimates*
  $r(s,a) \leftarrow 0$
  $n(s,a) \leftarrow 0$
  for all $s' \in S$ **do**
    $n(s,a,s') \leftarrow 0$
  end for
end for
for $t = 1,2,3,\cdots$ **do**
  Let s denote the state at time t.
  Choose action $a := \text{argmax}_{a' \in A} Q(s,a')$.
  Let r be the immediate reward and s' the next state after executing action a from state s.
  if $n(s,a) < m$ **then**
    $n(s,a) \leftarrow n(s,a) + 1$
    $r(s,a) \leftarrow r(s,a) + r$ // *Record immediate reward*
    $n(s,a,s') \leftarrow n(s,a,s') + 1$ // *Record immediate next-state*
    if $n(s,a) = m$ **then**
      for $i = 1,2,3,\cdots, \left\lceil \frac{\ln(1/(\varepsilon_1(1-\gamma)))}{1-\gamma} \right\rceil$ **do**
        for all $(\bar{s},\bar{a})$ **do**
          if $n(\bar{s},\bar{a}) \geq m$ **then**
            $Q(\bar{s},\bar{a}) \leftarrow \hat{R}(\bar{s},\bar{a}) + \gamma \sum_{s'} \hat{T}(s'|\bar{s},\bar{a}) \max_{a'} Q(s',a')$.
          end if

R-MAX

mean reward is

$$\hat{R}(s,a) := \frac{1}{n(s,a)} \sum_{i=1}^{n(s,a)} r[i].$$

Let $n(s,a,s')$ denote the number of times the agent has taken action a from state s and immediately transitioned to the state s'. Then, the *empirical transition distribution* is the distribution $\hat{T}(s,a)$ satisfying

$$\hat{T}(s'|s,a) := \frac{n(s,a,s')}{n(s,a)} \text{ for each } s' \in S.$$

In the R-MAX algorithm, the action-selection step is always to choose the action that maximizes the current action value, $Q(s,\cdot)$. The update step is to solve the following set of Bellman equations:

$$\begin{aligned} Q(s,a) &= \hat{R}(s,a) + \gamma \sum_{s'} \hat{T}(s'|s,a) \max_{a'} Q(s',a'), && \text{if } n(s,a) \geq m, \\ Q(s,a) &= U(s,a), && \text{otherwise,} \end{aligned} \tag{4}$$

where $\hat{R}(s,a)$ and $\hat{T}(\cdot|s,a)$ are the empirical (maximum-likelihood) estimates for the reward and transition distribution of state-action pair (s,a) using only data from the first m observations of (s,a). Solving this set of equations is equivalent to computing the optimal action-value function of an MDP, which we call *Model(R-MAX)*. This MDP uses the empirical transition and reward

R-MAX (Sample Complexity)

- Constructing known state-action pairs K_t

Let $n_t(s,a)$ denote the value of $n(s,a)$ at time t during execution of the algorithm. For R-MAX, let the "known" state-action pairs K_t, at time t (See Definition 6), to be

$$K_t := \{(s,a) \in \mathcal{S} \times \mathsf{A} | n_t(s,a) \geq m\},$$

- Determine event E

Event A1 *For all stationary policies* π, *timesteps* t *and states* s *during execution of the R-MAX algorithm on some MDP* M, $|V^{\pi}_{M_{K_t}}(s) - V^{\pi}_{\hat{M}_{K_t}}(s)| \leq \varepsilon_1$.

R-MAX (Sample Complexity)

- Determine parameter m

Lemma 15 *There exists a constant C such that if R-MAX with parameters m and ε_1 is executed on any MDP $M = \langle S, A, T, \mathcal{R}, \gamma \rangle$ and m satisfies*

$$m \geq C V_{\max}^2 \left(\frac{S + \ln(SA/\delta)}{{\varepsilon_1}^2 (1-\gamma)^2} \right) = \tilde{O}\left(\frac{S V_{\max}^2}{{\varepsilon_1}^2 (1-\gamma)^2} \right),$$

then Event A1 will occur with probability at least $1 - \delta$.

- Determine learning complexity

the learning complexity, $\zeta(\varepsilon, \delta) \leq |\{(s,a) | U(s,a) \geq V^*(s) - \varepsilon\}| m.$

R-MAX (Sample Complexity)

- Applying Theorem 10

Theorem 11 *Suppose that* $0 \leq \varepsilon < \frac{1}{1-\gamma}$ *and* $0 \leq \delta < 1$ *are two real numbers and* $M = \langle S, A, T, \mathcal{R}, \gamma \rangle$ *is any MDP. There exists inputs* $m = m(\frac{1}{\varepsilon}, \frac{1}{\delta})$ *and* ε_1, *satisfying* $m(\frac{1}{\varepsilon}, \frac{1}{\delta}) = O\left(\frac{(S+\ln(SA/\delta))V_{\max}^2}{\varepsilon^2(1-\gamma)^2}\right)$ *and* $\frac{1}{\varepsilon_1} = O(\frac{1}{\varepsilon})$, *such that if R-MAX is executed on M with inputs m and* ε_1, *then the following holds. Let* $\mathcal{A}_t$ *denote R-MAX's policy at time t and* s_t *denote the state at time t. With probability at least* $1-\delta$, $V_M^{\mathcal{A}_t}(s_t) \geq V_M^*(s_t) - \varepsilon$ *is true for all but*

$$O\left(\frac{|\{(s,a) \in S \times A | U(s,a) \geq V^*(s) - \varepsilon\}|}{\varepsilon^3(1-\gamma)^3}\left(S + \ln\frac{SA}{\delta}\right)V_{\max}^3 \ln\frac{1}{\delta}\ln\frac{1}{\varepsilon(1-\gamma)}\right)$$

timesteps t.

- Recall Theorem 10: $O\left(\frac{V_{\max}\zeta(\varepsilon,\delta)}{\varepsilon(1-\gamma)}\ln\frac{1}{\delta}\ln\frac{1}{\varepsilon(1-\gamma)}\right)$

R-MAX (Computing complexity)

On most timesteps, the R-MAX algorithm performs a constant amount of computation to choose its next action. Only when a state's last action has been tried m times does it solve its internal model. Our version of R-MAX uses value iteration to solve its model. Therefore, the per-timestep computational complexity of R-MAX is

$$\Theta\left(SA(S+\ln(A))\left(\frac{1}{1-\gamma}\right)\ln\frac{1}{\varepsilon_1(1-\gamma)}\right).$$

we see that the total computation time of R-MAX is $O\left(B+\frac{S^2A(S+\ln(A))}{1-\gamma}\ln\frac{1}{\varepsilon_1(1-\gamma)}\right)$

Delayed Q-learning

- Updating rule
  - when:

$$Q_t(s,a) - \left(\frac{1}{m} \sum_{i=1}^{m} (r_{k_i} + \gamma V_{k_i}(s_{k_i})) \right) \geq 2\varepsilon_1 .$$

  - update

$$Q_{t+1}(s,a) = \frac{1}{m} \sum_{i=1}^{m} (r_{k_i} + \gamma V_{k_i}(s_{k_i})) + \varepsilon_1$$

Algorithm 2 Delayed Q-learning

Inputs: $S, A, \gamma, m, \varepsilon_1$, and $U(\cdot,\cdot)$
for all (s,a) **do**
  $Q(s,a) \leftarrow U(s,a)$ *// action-value estimates*
  $AU(s,a) \leftarrow 0$ *// used for attempted updates*
  $l(s,a) \leftarrow 0$ *// counters*
  $b(s,a) \leftarrow 0$ *// beginning timestep of attempted update*
  $LEARN(s,a) \leftarrow true$ *// the LEARN flags*
end for
$t^* \leftarrow 0$ *// time of most recent action value change*
for $t = 1,2,3,\cdots$ **do**
  Let s denote the state at time t.
  Choose action $a := \text{argmax}_{a' \in A} Q(s,a')$.
  Let r be the immediate reward and s' the next state after executing action a from state s.
  if $b(s,a) \leq t^*$ **then**
    $LEARN(s,a) \leftarrow true$
  end if
  if $LEARN(s,a) = true$ **then**
    if $l(s,a) = 0$ **then**
      $b(s,a) \leftarrow t$
    end if
    $l(s,a) \leftarrow l(s,a) + 1$
    $AU(s,a) \leftarrow AU(s,a) + r + \gamma \max_{a'} Q(s',a')$
    if $l(s,a) = m$ **then**
      if $Q(s,a) - AU(s,a)/m \geq 2\varepsilon_1$ **then**
        $Q(s,a) \leftarrow AU(s,a)/m + \varepsilon_1$
        $t^* \leftarrow t$
      else if $b(s,a) > t^*$ **then**
        $LEARN(s,a) \leftarrow false$
      end if
      $AU(s,a) \leftarrow 0$
      $l(s,a) \leftarrow 0$

Delayed Q-learning

- Constructing known state-action pairs K_t

Definition 20 *During timestep t of the execution of Delayed Q-learning, we define K_t to be the set*

$$K_t := \left\{ (s,a) \in \mathcal{S} \times \mathrm{A} \mid Q_t(s,a) - \left(R(s,a) + \gamma \sum_{s'} T(s'|s,a) V_t(s') \right) \leq 3\varepsilon_1 \right\}.$$

- Defining event E

Definition 21 *Suppose we execute Delayed Q-learning in an MDP M. Define* **Event A2** *to be the event that for all timesteps t, if $(s,a) \notin K_{k_1}$ and an attempted update of (s,a) occurs during timestep t, then the update will be successful, where $k_1 < k_2 < \cdots < k_m = t$ are m last timesteps during which (s,a) is experienced consecutively by the agent.*

Delayed Q-learning

- Determine parameter m

Lemma 22 *Suppose we execute Delayed Q-learning with parameter m satisfying*

$$m \geq \frac{(1+\gamma V_{\max})^2}{2\varepsilon_1{}^2} \ln\left(\frac{3SA}{\delta}\left(1+\frac{SA}{\varepsilon_1(1-\gamma)}\right)\right)$$

in an MDP M. The probability that Event A2 occurs is greater than or equal to $1-\delta/3$.

- Defining event E

$$\zeta(\varepsilon,\delta) = O\left(2m\sum_{(s,a)\in S\times A}\frac{[U(s,a)-V^*(s)]_+}{\varepsilon_1}\right) = O\left(\frac{(1+\gamma V_{\max})^2\sum_{(s,a)\in S\times A}[U(s,a)-V^*(s)]_+}{\varepsilon^3(1-\gamma)^3}\ln\frac{SA}{\varepsilon\delta(1-\gamma)}\right)$$

Delayed Q-learning

- Applying Theorem 10

Theorem 16 *(Strehl et al., 2006b) Suppose that* $0 \leq \varepsilon < \frac{1}{1-\gamma}$ *and* $0 \leq \delta < 1$ *are two real numbers and* $M = \langle \mathcal{S}, \mathsf{A}, T, \mathcal{R}, \gamma \rangle$ *is any MDP. There exists inputs* $m = m(\frac{1}{\varepsilon}, \frac{1}{\delta})$ *and* ε_1*, satisfying* $m(\frac{1}{\varepsilon}, \frac{1}{\delta}) = O\left(\frac{(1+\gamma V_{\max})^2}{\varepsilon_1^2} \ln \frac{SA}{\varepsilon_1 \delta (1-\gamma)}\right)$ *and* $\frac{1}{\varepsilon_1} = O(\frac{1}{\varepsilon(1-\gamma)})$*, such that if Delayed Q-learning is executed on M, then the following holds. Let* $\mathcal{A}_t$ *denote Delayed Q-learning's policy at time t and* s_t *denote the state at time t. With probability at least* $1-\delta$*,* $V_M^{\mathcal{A}_t}(s_t) \geq V_M^*(s_t) - \varepsilon$ *is true for all but*

$$O\left(\frac{V_{\max}(1+\gamma V_{\max})^2 \sum_{(s,a)\in \mathcal{S}\times \mathsf{A}} [U(s,a) - V^*(s)]_+}{\varepsilon^4 (1-\gamma)^4} \ln\frac{1}{\delta} \ln\frac{1}{\varepsilon(1-\gamma)} \ln\frac{SA}{\delta\varepsilon(1-\gamma)}\right)$$

timesteps t.

Delayed Q-learing (Computing complexity)

On most timesteps, Delayed Q-learning performs only a constant amount of computation. Its worst-case computational complexity per timestep is

$$\Theta(\ln(A)),$$

for the current state. Since Delayed Q-learning performs at most $SA\left(1+\frac{SA}{(1-\gamma)\varepsilon_1}\right)$ attempted updates (see Lemma 19), each update involves m transitions, and each transition requires computing the greedy action whose computation complexity is $O(\ln(A))$, the total computation time of Delayed Q-learning is

$$O\left(B+\frac{mS^2A^2\ln(A)}{\varepsilon_1(1-\gamma)}\right),$$

A new lower bound

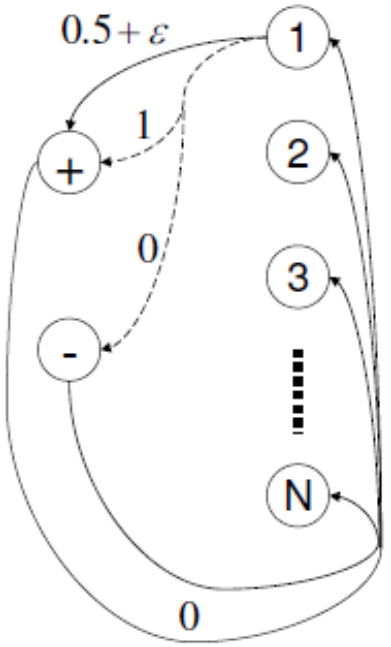
- Two assumptions:
  - $\mathcal{A}_t$ is a deterministic policy at all timesteps t, and
  - $\mathcal{A}_t$ and $\mathcal{A}_{t+1}$ can differ only in s_t; namely, the action-selection policy of the algorithm may change only in the most recently visited state.

Theorem 26 *For any reinforcement-learning algorithm $\mathcal{A}$ that satisfies the two assumptions above, there exists an MDP M such that the sample complexity of $\mathcal{A}$ in M is*

$$\Omega\left(\frac{SA}{\varepsilon^2}\ln\frac{S}{\delta}\right).$$

A new lower bound

- Building a difficult-to-learn MDP:



A new lower bound

Lemma 27 *There exist constants $c_1, c_2 \in (0,1)$ such that during a whole run of the algorithm $\mathcal{A}$, for any state $i \in [N]$, the probability that $\mathcal{A}$ takes sub-optimal actions in i more than m_i times is at least $p(m_i)$, where*

$$p(m_i) := c_2 \exp\left(-\frac{m_i \varepsilon^2}{c_1 A}\right).$$

Thanks

Kaiyuan Xu
xky@bu.edu

R-MAX (Sample Complexity)

Theorem 11 *Suppose that* $0 \leq \varepsilon < \frac{1}{1-\gamma}$ *and* $0 \leq \delta < 1$ *are two real numbers and* $M = \langle \mathcal{S}, \mathsf{A}, T, \mathcal{R}, \gamma \rangle$ *is any MDP. There exists inputs* $m = m(\frac{1}{\varepsilon}, \frac{1}{\delta})$ *and* ε_1*, satisfying* $m(\frac{1}{\varepsilon}, \frac{1}{\delta}) = O\left(\frac{(S+\ln(SA/\delta))V_{\max}^2}{\varepsilon^2(1-\gamma)^2}\right)$ *and* $\frac{1}{\varepsilon_1} = O(\frac{1}{\varepsilon})$*, such that if R-MAX is executed on* M *with inputs* m *and* ε_1*, then the following holds. Let* $\mathcal{A}_t$ *denote R-MAX's policy at time* t *and* s_t *denote the state at time* t*. With probability at least* $1-\delta$*,* $V_M^{\mathcal{A}_t}(s_t) \geq V_M^*(s_t) - \varepsilon$ *is true for all but*

$$O\left(\frac{|\{(s,a) \in \mathcal{S} \times \mathsf{A} | U(s,a) \geq V^*(s) - \varepsilon\}|}{\varepsilon^3(1-\gamma)^3}\left(S + \ln\frac{SA}{\delta}\right)V_{\max}^3 \ln\frac{1}{\delta}\ln\frac{1}{\varepsilon(1-\gamma)}\right)$$

timesteps t*.*

R-MAX (Sample Complexity)

Lemma 12 *(Strehl and Littman, 2005) Let* $M_1 = \langle S, \mathsf{A}, T_1, R_1, \gamma \rangle$ *and* $M_2 = \langle S, \mathsf{A}, T_2, R_2, \gamma \rangle$ *be two MDPs with non-negative rewards bounded by* 1 *and optimal value functions bounded by* $V_{\max}$. *Suppose that* $|R_1(s,a) - R_2(s,a)| \leq \alpha$ *and* $\|T_1(s,a,\cdot) - T_2(s,a,\cdot)\|_1 \leq 2\beta$ *for all states* s *and actions* a. *There exists a constant* $C > 0$ *such that for any* $0 \leq \varepsilon \leq 1/(1-\gamma)$ *and stationary policy* π, *if* $\alpha = 2\beta = C\varepsilon(1-\gamma)/V_{\max}$, *then*

$$|Q_1^{\pi}(s,a) - Q_2^{\pi}(s,a)| \leq \varepsilon.$$

Two Bounds

Lemma 13 *Suppose that* $r[1], r[2], \ldots, r[m]$ *are* m *rewards drawn independently from the reward distribution,* $\mathcal{R}(s,a)$*, for state-action pair* (s,a)*. Let* $\hat{R}(s,a)$ *be the empirical (maximum-likelihood) estimate of* $\mathcal{R}(s,a)$*. Let* δ_R *be any positive real number less than 1. Then, with probability at least* $1-\delta_R$*, we have that* $|\hat{R}(s,a) - \mathcal{R}(s,a)| \leq \varepsilon^R_{n(s,a)}$*, where*

$$\varepsilon^R_m := \sqrt{\frac{\ln(2/\delta_R)}{2m}}.$$

Proof This result follows directly from Hoeffding's bound (Hoeffding, 1963). ■

Lemma 14 *Suppose that* $\hat{T}(s,a)$ *is the empirical transition distribution for state-action pair* (s,a) *using* m *samples of next states drawn independently from the true transition distribution* $T(s,a)$*. Let* δ_T *be any positive real number less than 1. Then, with probability at least* $1-\delta_T$*, we have that* $\|T(s,a) - \hat{T}(s,a)\|_1 \leq \varepsilon^T_{n(s,a)}$ *where*

$$\varepsilon^T_m = \sqrt{\frac{2[\ln(2^S - 2) - \ln(\delta_T)]}{m}}.$$

R-MAX (Sample Complexity)

Lemma 15 *There exists a constant C such that if R-MAX with parameters m and* ε_1 *is executed on any MDP* $M = \langle S, A, T, \mathcal{R}, \gamma \rangle$ *and m satisfies*

$$m \geq CV_{\max}^2 \left(\frac{S + \ln(SA/\delta)}{\varepsilon_1{}^2 (1-\gamma)^2} \right) = \tilde{O}\left(\frac{SV_{\max}^2}{\varepsilon_1{}^2 (1-\gamma)^2} \right),$$

then Event A1 will occur with probability at least $1 - \delta$.

PAC-MDP Analysis Framework

Theorem 10 *Let $\mathcal{A}(\varepsilon,\delta)$ be any greedy learning algorithm such that, for every timestep t, there exists a set K_t of state-action pairs that depends only on the agent's history up to timestep t. We assume that $K_t = K_{t+1}$ unless, during timestep t, an update to some state-action value occurs or the escape event A_K happens. Let M_{K_t} be the known state-action MDP and π_t be the current greedy policy, that is, for all states s, $\pi_t(s) = \text{argmax}_a Q_t(s,a)$. Furthermore, assume $Q_t(s,a) \leq V_{\max}$ for all t and (s,a). Suppose that for any inputs ε and δ, with probability at least $1-\delta$, the following conditions hold for all states s, actions a, and timesteps t: (1) $V_t(s) \geq V^*(s) - \varepsilon$ (optimism), (2) $V_t(s) - V^{\pi_t}_{M_{K_t}}(s) \leq \varepsilon$ (accuracy), and (3) the total number of updates of action-value estimates plus the number of times the escape event from K_t, A_K, can occur is bounded by $\zeta(\varepsilon,\delta)$ (learning complexity). Then, when $\mathcal{A}(\varepsilon,\delta)$ is executed on any MDP M, it will follow a 4ε-optimal policy from its current state on all but*

$$O\left(\frac{V_{\max}\zeta(\varepsilon,\delta)}{\varepsilon(1-\gamma)}\ln\frac{1}{\delta}\ln\frac{1}{\varepsilon(1-\gamma)}\right)$$

timesteps, with probability at least $1-2\delta$.

R-MAX (Sample Complexity)

Theorem 11 *Suppose that* $0 \leq \varepsilon < \frac{1}{1-\gamma}$ *and* $0 \leq \delta < 1$ *are two real numbers and* $M = \langle \mathcal{S}, \mathsf{A}, T, \mathcal{R}, \gamma \rangle$ *is any MDP. There exists inputs* $m = m(\frac{1}{\varepsilon}, \frac{1}{\delta})$ *and* ε_1*, satisfying* $m(\frac{1}{\varepsilon}, \frac{1}{\delta}) = O\left(\frac{(S+\ln(SA/\delta))V_{\max}^2}{\varepsilon^2(1-\gamma)^2}\right)$ *and* $\frac{1}{\varepsilon_1} = O(\frac{1}{\varepsilon})$*, such that if R-MAX is executed on* M *with inputs* m *and* ε_1*, then the following holds. Let* $\mathcal{A}_t$ *denote R-MAX's policy at time* t *and* s_t *denote the state at time* t*. With probability at least* $1-\delta$*,* $V_M^{\mathcal{A}_t}(s_t) \geq V_M^*(s_t) - \varepsilon$ *is true for all but*

$$O\left(\frac{|\{(s,a) \in \mathcal{S} \times \mathsf{A} | U(s,a) \geq V^*(s) - \varepsilon\}|}{\varepsilon^3(1-\gamma)^3}\left(S + \ln\frac{SA}{\delta}\right)V_{\max}^3 \ln\frac{1}{\delta}\ln\frac{1}{\varepsilon(1-\gamma)}\right)$$

timesteps t*.*