RL Reading Group (III)

PAC-MDP

Kaiyuan Xu
xky@bu.edu

Reinforcement Learning in Finite MDPs: PAC Analysis

Alexander L. Strehl[*] ASTREHL@FACEBOOK.COM
Facebook
1601 S California Ave.
Palo Alto, CA 94304

Lihong Li[†] LIHONG@YAHOO-INC.COM
Yahoo! Research
4401 Great America Parkway
Santa Clara, CA 95054

Michael L. Littman MLITTMAN@CS.RUTGERS.EDU
Department of Computer Science
Rutgers University
Piscataway, NJ 08854

Abstract

We study the problem of learning near-optimal behavior in finite Markov Decision Processes (MDPs) with a polynomial number of samples. These "PAC-MDP" algorithms include the well-known E^3 and R-MAX algorithms as well as the more recent Delayed Q-learning algorithm. We summarize the current state-of-the-art by presenting bounds for the problem in a unified theoretical framework. A more refined analysis for upper and lower bounds is presented to yield insight into the differences between the model-free Delayed Q-learning and the model-based R-MAX.

Keywords: reinforcement learning, Markov decision processes, PAC-MDP, exploration, sample complexity

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

Sample Complexity (1.5, P2418)

Definition 1 *(Kakade 2003) Let* $c = (s_1, a_1, r_1, s_2, a_2, r_2, \ldots)$ *be a random path generated by executing an algorithm* $\mathcal{A}$ *in an MDP* M*. For any fixed* $\varepsilon > 0$*, the* **sample complexity of exploration** *(***sample complexity***, for short) of* $\mathcal{A}$ *is the number of timesteps* t *such that the policy at time* t*,* $\mathcal{A}_t$*, satisfies* $V^{\mathcal{A}_t}(s_t) < V^*(s_t) - \varepsilon$*.*

It directly measures the number of times the agent acts poorly

Main results

1.1 Main Results

We present two upper bounds and one lower bound on the achievable *sample complexity* of general reinforcement-learning algorithms (see Section 1.5 for a formal definition). The two upper bounds dominate all previously published bounds, but differ from one another. When logarithmic factors are ignored, the first bound, for the R-MAX algorithm, is

$$\tilde{O}(S^2A/(\varepsilon^3(1-\gamma)^6)),$$

while the corresponding second bound, for the Delayed Q-learning algorithm, is

$$\tilde{O}(SA/(\varepsilon^4(1-\gamma)^8)).$$

Based on the work of Mannor and Tsitsiklis (2004), we provide an improved lower bound

$$\Omega\left(\frac{SA}{\varepsilon^2}\ln\frac{S}{\delta}\right) \tag{3}$$

Notation and Some Assumptions

- MDP: <*S*, A, *T*, *R*, γ>
- R: S $\times$ A -> P_R, reward distribution
- T(s'|s, a): transition probability of state s' of the distribution T(s, a)
- $V_M^\pi(s) = \mathbf{E}[\sum_{j=1}^{\infty} \gamma^{j-1} r_j | s]$
- $V_M^\pi(s, H)$ denote the H-step value of policy π from s.

Notation and Some Assumptions

- The maximum reward is 1, thus V < 1/(1-γ)

1.3 Admissible Heuristics

We also assume that the algorithms are given an admissible heuristic for the problem before learning occurs. An **admissible heuristic** is a function $U : \mathcal{S} \times \mathsf{A} \to \mathbb{R}$ that satisfies $U(s,a) \geq Q^*(s,a)$ for all $s \in \mathcal{S}$ and $a \in \mathsf{A}$. We also assume that $U(s,a) \leq V_{\max}$ for all $(s,a) \in \mathcal{S} \times \mathsf{A}$ and some quantity $V_{\max}$. Prior information about the problem at hand can be encoded into the admissible heuristic and its upper bound $V_{\max}$. With no prior information, we can always set $U(s,a) = V_{\max} = 1/(1-\gamma)$ since $V^*(s) = \max_{a \in \mathsf{A}} Q^*(s,a)$ is at most $1/(1-\gamma)$. Therefore, without loss of generality, we assume $0 \leq U(s,a) \leq V_{\max} \leq 1/(1-\gamma)$ for all $(s,a) \in \mathcal{S} \times \mathsf{A}$.

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 $i = 1,2,3,\cdots, \boxed{\left\lceil \frac{\ln(1/(\varepsilon_1(1-\gamma)))}{1-\gamma} \right\rceil}$ **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

Quantifying iteration numbers

Proposition 3 *(Corollary 2 from Singh and Yee 1994) Let* $Q'(\cdot,\cdot)$ *and* $Q^*(\cdot,\cdot)$ *be two action-value functions over the same state and action spaces. Suppose that* Q^* *is the optimal value function of some MDP M. Let* π *be the greedy policy with respect to* Q' *and* π^* *be the greedy policy with respect to* Q^**, which is the optimal policy for M. For any* $\alpha > 0$ *and discount factor* $\gamma < 1$*, if* $\max_{s,a}\{|Q'(s,a) - Q^*(s,a)|\} \leq \alpha(1-\gamma)/2$*, then* $\max_s\{V^{\pi^*}(s) - V^{\pi}(s)\} \leq \alpha$.

Proposition 4 *Let* $\beta > 0$ *be any real number satisfying* $\beta < 1/(1-\gamma)$ *where* $\gamma < 1$ *is the discount factor. Suppose that value iteration is run for* $\left\lceil \frac{\ln(1/(\beta(1-\gamma)))}{1-\gamma} \right\rceil$ *iterations where each initial action-value estimate,* $Q(\cdot,\cdot)$*, is initialized to some value between* 0 *and* $1/(1-\gamma)$*. Let* $Q'(\cdot,\cdot)$ *be the resulting action-value estimates. Then, we have that* $\max_{s,a}\{|Q'(s,a) - Q^*(s,a)|\} \leq \beta$.

PAC-MDP Analysis

3.1 General Framework

We now develop some theoretical machinery to prove PAC-MDP statements about various algorithms. Our theory will be focused on algorithms that maintain a table of action values, $Q(s,a)$, for each state-action pair (denoted $Q_t(s,a)$ at time t).[10] We also assume an algorithm always chooses actions greedily with respect to the action values. This constraint is not really a restriction, since we could define an algorithm's action values as 1 for the action it chooses and 0 for all other actions. However, the general framework is understood and developed more easily under the above assumptions. For convenience, we also introduce the notation $V(s)$ to denote $\max_a Q(s,a)$ and $V_t(s)$ to denote $V(s)$ at time t.

Definition 5 *Suppose an RL algorithm $\mathcal{A}$ maintains a value, denoted $Q(s,a)$, for each state-action pair $(s,a) \in \mathcal{S} \times \mathsf{A}$. Let $Q_t(s,a)$ denote the estimate for (s,a) immediately before the t^{th} action of the agent. We say that $\mathcal{A}$ is a* **greedy algorithm** *if the t^{th} action of $\mathcal{A}$, a_t, is $a_t := \operatorname{argmax}_{a \in \mathsf{A}} Q_t(s_t, a)$, where s_t is the t^{th} state reached by the agent.*

PAC-MDP Analysis

Definition 6 *Let $M = \langle \mathcal{S}, \mathsf{A}, T, \mathcal{R}, \gamma \rangle$ be an MDP with a given set of action values, $Q(s,a)$, for each state-action pair (s,a), and a set K of state-action pairs, called the* **known state-action pairs**. *We define the* **known state-action MDP** $M_K = \langle \mathcal{S} \cup \{z_{s,a} | (s,a) \notin K\}, \mathsf{A}, T_K, R_K, \gamma \rangle$ *as follows. For each unknown state-action pair, $(s,a) \notin K$, we add a new state $z_{s,a}$ to M_K, which has self-loops for each action ($T_K(z_{s,a}|z_{s,a},\cdot) = 1$). For all $(s,a) \in K$, $R_K(s,a) = R(s,a)$ and $T_K(\cdot|s,a) = T(\cdot|s,a)$. For all $(s,a) \notin K$, $R_K(s,a) = Q(s,a)(1-\gamma)$ and $T_K(z_{s,a}|s,a) = 1$. For the new states, the reward is $R_K(z_{s,a},\cdot) = Q(s,a)(1-\gamma)$.*

Definition 7 *For algorithm $\mathcal{A}$, for each timestep t, let K_t (we drop the subscript t if t is clear from context) be a set of state-action pairs defined arbitrarily in a way that depends only on the history of the agent up to timestep t (before the $(t)^{\text{th}}$ action). We define A_K to be the event, called the* **escape event**, *that some state-action pair $(s,a) \notin K_t$ is experienced by the agent at time t.*

Some Bounds

Chernoff-Hoeffding Bound

Lemma 8 *Suppose a weighted coin, when flipped, has probability $p > 0$ of landing with heads up. Then, for any positive integer k and real number $\delta \in (0,1)$, there exists a number $m = O((k/p)\ln(1/\delta))$, such that after m tosses, with probability at least $1-\delta$, we will observe k or more heads.*

we assume $V^*_M(s) \leq V_{\max}$ and $Q(s,a) \leq V_{\max}$ for all $s \in \mathcal{S}$ and $a \in \mathsf{A}$.

Lemma 9 *Let $M = \langle \mathcal{S}, \mathsf{A}, T, \mathcal{R}, \gamma \rangle$ be an MDP whose optimal value function is upper bounded by $V_{\max}$. Furthermore, let M_K be a known state-action MDP for some $K \subseteq \mathcal{S} \times \mathsf{A}$ defined using value function $Q(s,a)$. Then, $V^*_{M_K}(s) \leq V_{\max} + \max_{s',a'} Q(s',a')$ for all $s \in \mathcal{S}$.*

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, $\pi_t(s) = \mathrm{argmax}_a Q_t(s,a)$. Furthermore, assume $Q_t(s,a) \le V_{\max}$ 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) $V_t(s) \ge V^*(s) - \varepsilon$ (optimism), (2) $V_t(s) - V^{\pi_t}_{M_{K_t}}(s) \le \varepsilon$ (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 $1-2\delta$.

Proof.

$$
\begin{aligned}
&V_M^{\mathcal{A}_t}(s_t, H) \\
&\quad \geq V_{M_{K_t}}^{\pi_t}(s_t, H) - 2V_{\max} \Pr(W) \\
&\quad \geq V_{M_{K_t}}^{\pi_t}(s_t) - \varepsilon - 2V_{\max} \Pr(W) \\
&\quad \geq V(s_t) - 2\varepsilon - 2V_{\max} \Pr(W) \\
&\quad \geq V^*(s_t) - 3\varepsilon - 2V_{\max} \Pr(W).
\end{aligned}
$$

The first step above follows from the fact that following $\mathcal{A}_t$ in MDP M results in behavior identical to that of following π_t in M_{K_t} unless event W occurs, in which case a loss of at most $2V_{\max}$ can occur (Lemma 9). The second step follows from the definition of H above. The third and final steps follow from Conditions 2 and 1, respectively, of the proposition.

Proof.

Now, suppose that $\Pr(W) < \frac{\varepsilon}{2V_{\max}}$. Then, we have that the agent's policy on timestep t is 4ε-optimal:

$$V_M^{\mathcal{A}_t}(s_t) \geq V_M^{\mathcal{A}_t}(s_t, H) \geq V_M^*(s_t) - 4\varepsilon.$$

Otherwise, we have that $\Pr(W) \geq \frac{\varepsilon}{2V_{\max}}$, which implies that an agent following $\mathcal{A}_t$ will either perform a successful update in H timesteps, or encounter some $(s,a) \notin K_t$ in H timesteps, with probability at least $\frac{\varepsilon}{2V_{\max}}$. Call such an event a "success". Then, by Lemma 8, after $O(\frac{\zeta(\varepsilon,\delta)HV_{\max}}{\varepsilon}\ln 1/\delta)$ timesteps t where $\Pr(W) \geq \frac{\varepsilon}{2V_{\max}}$, $\zeta(\varepsilon,\delta)$ successes will occur, with probability at least $1-\delta$. Here, we have identified the event that a success occurs after following the agent's policy for H steps with the event that a coin lands with heads facing up. However, by Condition 3 of the proposition, with probability at least $1-\delta$, $\zeta(\varepsilon,\delta)$ is the maximum number of successes that will occur throughout the execution of the algorithm.

To summarize, we have shown that with probability $1-2\delta$, the agent will execute a 4ε-optimal policy on all but $O(\frac{\zeta(\varepsilon,\delta)HV_{\max}}{\varepsilon}\ln\frac{1}{\delta}) = O(\frac{\zeta(\varepsilon,\delta)V_{\max}}{\varepsilon(1-\gamma)}\ln\frac{1}{\delta}\ln\frac{1}{\varepsilon(1-\gamma)})$ timesteps. ■

Thanks

Kaiyuan Xu
xky@bu.edu

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

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* $M_1 = \langle \mathcal{S}, \mathsf{A}, T_1, R_1, \gamma \rangle$ *and* $M_2 = \langle \mathcal{S}, \mathsf{A}, T_2, R_2, \gamma \rangle$ *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* $M = \langle \mathcal{S}, \mathsf{A}, T, \mathcal{R}, \gamma \rangle$ *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* $1 - \delta$.

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*,* $\pi_t(s) = \mathrm{argmax}_a Q_t(s, a)$*. 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.