Model-based reinforcement learning with nearly tight exploration complexity bounds

István Szita SZITYU@GMAIL.COM
Csaba Szepesvári SZEPESVA@CS.UALBERTA.CA
University of Alberta, Athabasca Hall, Edmonton, AB T6G 2E8 Canada

Abstract

One might believe that model-based algorithms of reinforcement learning can propagate the obtained experience more quickly, and are able to direct exploration better. As a consequence, fewer exploratory actions should be enough to learn a good policy. Strangely enough, current theoretical results for model-based algorithms do not support this claim: In a finite Markov decision process with N states, the best bounds on the number of exploratory steps necessary are of order $O(N^2 \log N)$, in contrast to the $O(N \log N)$ bound available for the model-free, DELAYED Q-LEARNING algorithm. In this paper we show that MORMAX, a modified version of the RMAX algorithm needs to make at most $O(N \log N)$ exploratory steps. This matches the lower bound up to logarithmic factors, as well as the upper bound of the state-of-the-art model-free algorithm, while our new bound improves the dependence on other problem parameters.

In the reinforcement learning (RL) framework, an agent interacts with an unknown environment and tries to maximize its long-term profit. A standard way to measure the efficiency of the agent is *sample complexity* or *exploration complexity*. Roughly, this quantity tells how many non-optimal (exploratory) steps does the agent make at most.

The best understood and most studied case is when the environment is a finite Markov decision process (MDP) with the expected total discounted reward criterion. Since the work of Kearns & Singh (1998), many algorithms have been published with bounds on their sample complexity. Some of these algorithms like RMAX, MBIE, OIM build an approximate model of the environment, while others, like DELAYED Q-LEARNING, do not, but rather approximate an action value function directly.

Intuitively, keeping a model should help: whenever some information is gained from an exploratory move, a model-based algorithm can immediately propagate that information throughout the state space. More efficient use of information means that, at least in principle, less exploration is needed.

In this light, it seems surprising that the best known sample complexity bounds are better for model-free methods than for the model-based ones: the sample complexity of DELAYED Q-LEARNING scales linearly with the number of states, while the previously best known bounds for model-based methods scale quadratically. One may speculate that this is because in a model the number of transitions that need to be stored (and thus estimated) is quadratic in the number of states.

Here we show that this is not the case: The quadratic bounds of earlier results were *not* due to the inherent limitations of the model-based approach. Specifically, we present a slight modification of the RMAX algorithm and show that (disregarding logarithmic factors) its sample complexity scales only linearly with the number of states, and has better dependence on the value upper bound $V_{\max}$ and the required precision ϵ than the previous best bound, which was available for a model-free algorithm.

1. Markov decision processes

A standard assumption of RL is that the environment is (discounted-reward, finite) MDP. Here we only introduce the notational framework used in the paper, for detailed explanations we refer the reader to Bertsekas & Tsitsiklis (1996). Technically, a finite MDP M

Appearing in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010.

is a triple $(X, A, \mathcal{P})$, where X is a finite set of states; A is a finite set of possible actions; and $\mathcal{P}$, determining the evolution of the decision process, maps $X \times A$ into distributions over $X \times \mathbb{R}$ (states and rewards). We assume that all the (random) immediate rewards are nonnegative and are upper-bounded by a constant $R_{\max} \geq 0$. By slightly abusing the notation when the stochasticity of the rewards is not a concern, we can identify M with the 4-tuple (X, A, P, R) for an MDP, where $P : X \times A \times X$ is the transition kernel and $R : X \times A \to [0, R_{\max}]$ is the immediate expected reward function underlying $\mathcal{P}$.

A policy π is a mapping that assigns to every history h a probability mass function over the actions A. Following a policy in the MDP means that $a_t \sim \pi(\cdot|h_t)$. A stationary policy is identified with a mapping $\pi : X \times A \to [0, 1]$. We assume that future rewards are discounted by a multiplicative factor $\gamma \in [0, 1)$. The discounted state- and action-value functions will be denoted by V^π and Q^π, respectively, for any (not necessarily stationary) policy π. The optimal state- and action-value functions will be denoted by V^* and Q^*.

If the parameters of the MDP, P and R are known, then the optimal value function (and thus the optimal policy) can be found by an iterative solution of the Bellman-equations (Bertsekas & Tsitsiklis, 1996). In the reinforcement learning setting, however, P and R are unknown, and the agent has only access to direct experience: being in state x_t and taking action a_t, it can observe the next state x_{t+1} and the reward r_t. For simplicity, in this paper we assume that the reward function is known, while the transition probabilities are not.[1]

Exploration. In an unknown environment, the agent has to make exploratory actions to learn the effects of its actions, and thus learn how to get to the most rewarding parts of the state space. This means that from time to time, the agent has to take actions other than the ones that seem best for the moment, in the hope that it finds something better. Failing to do that, the agent may never find the optimal policy. On the other hand, too much exploration reduces the amount of collected rewards unnecessarily. Finding the right balance is non-trivial.

Model-based and model-free learning. There are three main branches of RL methods for learning in MDPs. Of course, the boundaries of these three categories are somewhat blurred.

Model-based methods approximate the transition probabilities and the reward function, then derive a value function from the approximate MDP. The policy of the agent is then derived from the value function. *Model-free methods* directly learn a value function (and from that, a policy). The third branch consists of *policy search* algorithms which aim at finding a policy without resorting to estimating value functions. As no sample complexity results are known about policy search methods, we will not discuss them in this paper.

[1] The results would continue to hold in the more general case with some obvious modifications.

1.1. Sample complexity of reinforcement learning

One natural evaluation criterion for RL algorithms is to count the number of non-optimal actions taken (these are the exploratory actions or those that the agent mistakenly believes to be optimal). Before giving the precise definition, we need the concept of MDP learning algorithms. An **MDP learning algorithm** (in short, algorithm) is a procedure which gets as input the number of states, actions, γ, $R_{\max}$ (and possibly other parameters) of some MDP M and then interacts with M by sequentially observing states and rewards and sending actions to M. Actions must still be chosen based on past observations. Given $x_0, a_0, r_0, x_1, a_1, r_1, \ldots$, a stream of experience generated when algorithm $\mathcal{A}$ interacts with M, we define its (expected future) value at time t, conditioned on the past h_t as $V^{\mathcal{A}}_{t,M} = \mathbb{E}\left[\sum_{s=0}^{\infty} \gamma^s r_{t+s} \,|\, h_t\right]$. Note that $V^{\mathcal{A}}_{t,M}$ is a random variable. Let $V_{\max}$ be an upper bound on the state values in the MDP. (Clearly, we can choose $V_{\max} = R_{\max}/(1-\gamma)$, but sometimes much better bounds are available.)

Definition 1 (Kakade, 2003) *Let $\epsilon > 0$ be a prescribed accuracy and $\delta > 0$ be an allowed probability of failure. The expression $\zeta(\epsilon, \delta, N, K, \gamma, R_{\max})$ is a sample complexity bound for algorithm $\mathcal{A}$, if the following holds: Take any $\epsilon > 0, \delta \in (0, 1)$, $N > 0$, $K > 0$, $\gamma \in [0, 1)$, $R_{\max} > 0$ and any MDP M with N states, K actions, discount factor γ, and rewards bounded by $R_{\max}$. Let $\mathcal{A}$ interact with M, resulting in the process $x_0, a_0, r_0, x_1, a_1, r_1, \ldots$. Then, independently of the choice of x_0, with probability at least $1 - \delta$, the number of timesteps such that $V^{\mathcal{A}}_{t,M} < V^*(x_t) - \epsilon$ is at most $\zeta(\epsilon, \delta, N, K, \gamma, R_{\max})$.*[2]

An algorithm with sample complexity that is polynomial in $1/\varepsilon$, $\log(1/\delta)$, N, K, $1/(1-\gamma)$, $R_{\max}$ is

[2] An alternative way for measuring sample complexity is to count the number of timesteps when $Q^*(x_t, a_t) < V^*(x_t) - \epsilon$. If ζ' is a uniform bound on this count, and ζ is as above then one can show that $\zeta' \leq \zeta$.

called *PAC-MDP* (probably approximately correct in MDPs).

1.2. Previous sample complexity results

The above definition of sample complexity is due to Kakade (2003), though algorithms with polynomial sample complexity existed before. The first such algorithm was E^3 (Kearns & Singh, 2002), which was followed by other, simpler algorithms like RMAX (Brafman & Tennenholtz, 2002), MODEL-BASED INTERVAL-ESTIMATION (MBIE) (Strehl & Littman, 2005) and OPTIMISTIC INITIAL MODEL (OIM) (Szita & Lőrincz, 2008), all of which are model-based. The state-of-the-art complexity bound for model-based learning in MDPs is due to Li (2009), who showed that RMAX takes at most $\widetilde{O}\left(\frac{N^2KV_{\max}^3}{\epsilon^3(1-\gamma)^3}\right)$ exploratory actions.[3] The only model-free method with known complexity bounds is for DELAYED Q-LEARNING (Strehl et al., 2006), which requires at most $\widetilde{O}\left(\frac{NKV_{\max}^4}{\epsilon^4(1-\gamma)^4}\right)$ exploratory actions.

It is worth mentioning that if the agent has access to an oracle that can draw samples from $P(x,a,y)$ for any (x,a) pair, then $\widetilde{O}\left(\frac{NKV_{\max}^2}{\epsilon^2(1-\gamma)^2}\right)$ samples are sufficient (Kearns & Singh, 1998). The best available lower bound is by Li (2009): at least $\Omega(\frac{NKR_{\max}^2}{\epsilon^2}\log\frac{N}{\delta})$ exploratory moves are necessary for any deterministic algorithm that learns from samples.

An alternative interesting measure for the effectiveness of an RL algorithm is *regret*, the total reward lost by the algorithm in comparison to the optimal policy (e.g., Auer et al. 2009). Discussion of regret bounds is out of the scope of the present paper.

2. The modified RMAX algorithm

The original RMAX algorithm is an archetypical example of a model-based RL algorithm. The model of RMAX is not accurate (especially not at the beginning of learning when the agent has basically no information about the environment), but it ensures that whenever there is uncertainty about a state, its value will be overestimated. Initially, the model assumes that each state is worth $R_{\max}$ immediate reward (hence the name), and all actions self-loop, thus their value is $V_{\max} = R_{\max}/(1-\gamma)$. As soon as enough samples are collected about a state, the agent estimates the transition probabilities (and rewards) from that state, then recalculates the optimal policy according to the current model. As the value of unknown states is overestimated, the agent will be drawn towards them until they are explored.

The original algorithm needs $m = O(N\log N)$ samples (considering only N-dependency) per state-action (SA) pairs. After collecting this many samples for an SA-pair, RMAX considers the pair known, and stops collecting further information. The analysis of DELAYED Q-LEARNING (Strehl et al., 2006) and parallel sampling (Kearns & Singh, 1998) suggests that $m = O(\log N)$ samples might be enough if we keep collecting fresh data, and when an m-element batch of new data is collected, we re-estimate the model if necessary. Obviously, if we want any benefits, the number of re-estimations cannot be too large. Let t be the current time step and $\widehat{\tau}_{t,x,a}$ be the last time prior to t when an update of the model at (x,a) was considered (or 0 if no update happened yet at (x,a)). We will show that no update is necessary, if no updates happened elsewhere in the model between $\widehat{\tau}_{t,x,a}$ and t. Nor is an update necessary unless the value of a SA-pair is decreased significantly, say, by $c\cdot\epsilon$ with some appropriate constant $c>0$. Consequently, any (x,a) pair will be updated at most $V_{\max}/(c\epsilon)$ times, which is affordable. The resulting MORMAX algorithm is listed as Alg. 1.[4] To increase readability, we omitted subscripts. To match with the quantities used in the proofs, set $a\leftarrow a_t$, $x\leftarrow x_t$, $y\leftarrow x_{t+1}$. Other than that, all left-hand side quantities should be assigned index $t+1$, right-hand side ones index t. One exception is $\widehat{Q}_{prev}(x,a)$, which refers to $\widehat{Q}_{\widehat{\tau}_{x,a}}(x,a)$. Note that if we simplify the condition in line 14 to 'if $(\widehat{\tau}_{x,a}=0)$ then…' and increase m, we get back the original RMAX algorithm.

We note that the naive representation of $\widehat{P}$ requires $O(N^2K)$ space. Using a more effective implementation, such as the one used by Kearns & Singh (1998), does not store the full transition probability table, only the $m=O(\log(NK))$ samples for each SA-pair, so the total space complexity can be cut down to $\widetilde{O}(NK)$.

Theorem 1 *Fix some prescribed accuracy $\epsilon>0$, failure probability $\delta\in(0,1)$, and discount factor $\gamma\in[0,1)$. Let $M=(X,A,P,R)$ be an MDP with $|X|=N$ states and $|A|=K$ actions, with nonnegative rewards, and a value $V_{\max}\in\mathbb{R}$ that is an upper bound on all discounted cumulated rewards. If the* MORMAX *algorithm runs on MDP M then with probability at least $1-\delta$, the number of time steps t for which $V_{M,t}^{\text{MORMAX}} < V_M^*(x)-\epsilon$ is bounded by $\widetilde{O}\left(\frac{NKV_{\max}^2}{(1-\gamma)^4\epsilon^2}\right)$.*

[3] The notation $\widetilde{O}(n)$ stands for $O(n\log n)$.

[4] MORMAX stands for MODIFIED RMAX.

Algorithm 1 The MORMAX algorithm

Input: discount γ, accuracy ϵ, allowed failure prob. δ, value upper bound $V_{\max}$
Initialization:
$t := 0$; $\tau^* := 0$; $\epsilon_2 := (1-\gamma)\epsilon/(15\gamma)$; $m := \frac{1800\gamma^2 V_{\max}^2}{\epsilon^2(1-\gamma)^2} \log \frac{144\gamma N^2 K^2 V_{\max}}{\delta\epsilon(1-\gamma)}$
for all $(x,a,y) \in X \times A \times X$ **do**
  $n(x,a,y) := 0$; $n(x,a) := 0$ $\widehat{P}(x,a,y) := \mathbb{I}\{x=y\}$; $\widehat{R}(x,a) := (1-\gamma)V_{\max}$; $\widehat{\tau}_{x,a} := 0$;
$\widehat{Q} := \texttt{SolveMDP}(\widehat{P}, \widehat{R}, \gamma)$
Receive x from the environment
repeat
  $a := \arg\max_{a'} \widehat{Q}(x,a')$ *//greedy action w.r.t. current optimistic model*
  interact: Apply a and observe r and y
  $n(x,a) := n(x,a)+1$; $n(x,a,y) := n(x,a,y)+1$ *// update counters*
  if $n(x,a) = m$ **then**
    // if (x,a) is not yet known or change is significant, we update the model
    if $(\widehat{\tau}_{x,a} = 0) \vee [(\tau^* > \widehat{\tau}_{x,a}) \wedge (\widehat{Q}_{prev}(x,a) - \widehat{Q}(x,a) > \gamma\epsilon_2)]$ **then**
      $P' := \widehat{P}$, $R' := \widehat{R}$; $P'(x,a,\cdot) = n(x,a,\cdot)/m$; $R'(x,a) = R(x,a)$
      $Q' = \texttt{SolveMDP}(P', R', \gamma)$ *// calculate trial model*
      if $Q'(x,a) \le \widehat{Q}(x,a)$ **then**
        $\widehat{P} := P'$; $\widehat{R} := R'$; $\widehat{Q} := Q'$ *// model update*
      $\widehat{\tau}_{x,a} := t$; $\tau^* := t$; $\widehat{Q}_{prev}(x,a) := \widehat{Q}(x,a)$
    $n(x,a) := 0$, $n(x,a,\cdot) := 0$ *// restart sample collection at (x,a)*
  $t := t+1$; $x := y$
until interaction lasts

3. Bounding sample complexity

For a reader familiar with RL sample complexity bounds, most of the following proof may look familiar: the logic of the proof is fairly standard. Elements of the proof appear in the works of Kearns & Singh (1998); Kakade (2003); Szita & Lőrincz (2008); Li (2009) and Strehl et al. (2006). Nevertheless, we provide a detailed sketch of the proof because our improvement on the key lemma induces changes in all other parts of the proof, so citing the original versions of various lemmas from the abovementioned papers would inevitably lead to confusion and loss of rigour. For lack of space, easy and/or well-known results are stated as lemmas without proofs.

3.1. Definitions

Let us fix $\epsilon_1 = O(\epsilon)$ and $\delta_1 = O(\delta)$, with their exact values defined later. For any time step t, let x_t be the actual state and a_t be the action of the agent; and let $\mathcal{F}_t$ be the σ-algebra defined by the history up to t, that is, $\mathcal{F}_t = \sigma\{x_0, a_0, r_0, \ldots, x_t, a_t, r_t, x_{t+1}\}$. [5]

Update times, phase end times, known SA-pairs. Given $t \ge 1$, we define the phase end times $\tau_{t,x,a}$ as the last time step before t when m new samples for (x,a) were collected. Furthermore, we have already defined the update times $\widehat{\tau}_{t,x,a}$ as the last time a model update was considered. A model update is considered whenever $\widehat{Q}_t(x,a)$ has decreased significantly since the last update time. It generally means that the model is updated—unless that would cause an increase of $\widehat{Q}_t$ (in this case only $\widehat{\tau}$ is updated). Let K_t be the set of SA-pairs that are "known" at time t, that is, where the agent has a reasonable approximation to $P(x,a,\cdot)$: $K_t \stackrel{\text{def}}{=} \{(x,a) \in X \times A : \widehat{\tau}_{t,x,a} > 0\}$.

Approximate model. The approximate model $\widehat{M}_t = (X, A, \widehat{P}_t, \widehat{R}_t)$ is defined with

$$\widehat{P}_t(x,a,y) \stackrel{\text{def}}{=} \begin{cases} n_{\widehat{\tau}_{t,x,a}}(x,a,y)/m, & \text{if } (x,a) \in K_t; \\ \mathbb{I}\{x=y\}, & \text{if } (x,a) \notin K_t, \end{cases}^{6}$$

$$\widehat{R}_t(x,a) \stackrel{\text{def}}{=} \begin{cases} R(x,a), & \text{if } (x,a) \in K_t; \\ R_{\max}, & \text{if } (x,a) \notin K_t. \end{cases}$$

Let $\widehat{\pi}_t$ be the optimal policy of the MDP $\widehat{M}_t$ and $Q^{\widehat{\pi}_t}_{\widehat{M}_t}$ be its action-value function.

Truncation, ϵ_1-horizon. Fix an MDP M, discount factor γ, and a (not necessarily memoryless) policy π. Let $\mathcal{R}_0, \mathcal{R}_1, \ldots$ be the random sequence of re-

[5] Readers not experienced with measure theory can read "quantity X is $\mathcal{F}_t$ measurable" as "if we know the history up to t, then we can also compute quantity X".

[6] $\mathbb{I}\{\cdot\}$ is the indicator function.

wards generated by M and π when the initial state is chosen uniformly at random from X. For any $H \in \mathbb{N}$ and $x \in X$, define the H-step truncated value $V_M^\pi(x;H) \stackrel{\text{def}}{=} \mathbb{E}\left[\sum_{i=0}^{H-1} \gamma^i \mathcal{R}_i | x_0 = x\right]$. Define the effective ϵ_1-horizon $H = H(\epsilon_1) \stackrel{\text{def}}{=} \frac{1}{1-\gamma} \ln \frac{V_{\max}}{\epsilon_1}$.

The "nice" set. Let $\epsilon_2 = \epsilon(1-\gamma)/(15\gamma)$ and let L_t be the set of SA-pairs where the effect of using the approximate model is small:

$$L_t \stackrel{\text{def}}{=} \{(x,a) \in K_t : \sum_{y \in X} (\widehat{P}_t - P)(x,a,y) V_{\widehat{M}_t}^{\widehat{\pi}_t}(y) \le 3\epsilon_2\}.$$

Note that all nice SA-pairs are known, but the set can be much smaller: we will prove later that (x,a) is nice if no model update is likely to happen in H steps, if the agent starts form (x,a) with its current knowledge. The Idea of nice sets appears first in the analysis of delayed Q-learning (Strehl et al., 2006), but in our case, the situation is more complex, because model updates in a single (x,a) may affect the whole state space.

Escape event. Define E_t as the event that the approximate model of the agent will change within the next H steps, either because a state becomes known or because the transition probabilities of an SA-pair are updated. Formally,

$$E_t \stackrel{\text{def}}{=} \bigcup_{i=0}^{H-1} \{\widehat{Q}_{t+i+1} \neq \widehat{Q}_{t+i}\} \cup \{x_{t+i} \notin L_{t+i}\}.$$

Let e_t be the conditional probability that E_t happens: $e_t \stackrel{\text{def}}{=} \mathbb{P}(E_t | \mathcal{F}_t)$.

Auxiliary MDP. We define an MDP $\overline{M}_t = (X, A, \overline{P}_t, \overline{R}_t)$ which is the same as M on L_t; all states in $X \setminus K_t$ are absorbing with value $V_{\max}$, and the states in $K_t \setminus L_t$ are also absorbing, but have the same value as in $\widehat{M}_t$ for policy $\widehat{\pi}_t$:

$$\overline{P}_t(x,a,y) \stackrel{\text{def}}{=} \begin{cases} P(x,a,y), & \text{if } (x,a) \in L_t; \\ \mathbb{I}\{x=y\}, & \text{otherwise,} \end{cases}$$

$$\overline{R}_t(x,a) \stackrel{\text{def}}{=} \begin{cases} R(x,a), & \text{if } (x,a) \in L_t; \\ (1-\gamma)\widehat{Q}_t(x,a), & \text{otherwise.} \end{cases}$$

Note that the agent does not know $\overline{M}_t$ (as it does not know the true transition probabilities and L_t), but $\overline{M}_t$ will be useful in the proof to connect policy values in $\widehat{M}_t$ and M.

3.2. The fundamental approximation lemma

We begin the proof with a lemma that bounds the distance of the empirical transition probability estimates to their true values. This lemma is responsible for determining the complexity bound of the algorithm.

Lemma 2 *Fix $\delta \in (0,1)$, $\epsilon > 0$, $V_{\max} > 0$, $m \in \mathbb{N}$, an arbitrary MDP $M = (X, A, P, R)$, and a (possibly history dependent) policy π of M. Let $x_0, a_0, r_0, x_1, a_1, r_1, \ldots, x_{t-1}, a_{t-1}, r_{t-1}, x_t, \ldots$ be the sequence of state-action-reward triplets generated by following π in M from a random initial state x_0. Let $\mathcal{F}_t = \sigma\{x_0, a_0, r_1 \ldots, x_t, a_t, r_{t+1}, x_{t+1}\}$, and τ be a stopping time w.r.t. $\{\mathcal{F}_t\}$. Fix $(x,a) \in X \times A$ and let t_k be the k^{th} time step such that $t \ge \tau$ and $(x_t, a_t) = (x,a)$ (possibly infinity). Let $V : X \to \mathbb{R}$ be any $\mathcal{F}_\tau$-measurable value function satisfying $0 \le V(z) \le V_{\max}$ for all $z \in X$. Define the approximate transition probabilities for all $y \in X$:*

$$\widehat{P}^\infty(y) \stackrel{\text{def}}{=} \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}\{t_i < \infty; x_{t_i+1} = y\} + \mathbb{I}(t_i = \infty) P(x,a,y).$$

If $m \ge \frac{8V_{\max}^2}{\epsilon^2} \log \frac{2}{\delta}$ then with probability $1-\delta$, $\left|\sum_{y \in X} (P(x,a,y) - \widehat{P}^\infty(y)) V(y)\right| < \epsilon$.

Similar versions of the lemma can be found in the related literature, e.g. Lemma 9.1.1. of Kakade (2003). There are two significant differences: (1) Previous variants of the lemma did not consider the possibility $t_i = \infty$ (which may happen with nonzero probability, for example, if the algorithm decides that no path containing x can be optimal, so it stops visiting it). Therefore, it is not correct to apply the previous variants unless under all policies the visit probabilities of all states are uniformly bounded away from zero. (2) We allow the value function V to be random as long as it is $\mathcal{F}_\tau$-measurable. Therefore, we do not need to get a good approximation on all possible instances of the N-dimensional space of value functions, only the polynomially many (random) value functions V_t that we encounter during the run of the algorithm. This allows us to use fewer samples, but the improvement comes at a price: we need to make some changes to the algorithm itself so that the lemma becomes applicable. So, understanding the assumptions of the lemma is necessary to understand why the changes in the algorithm are necessary.

3.3. The number of non-optimal steps

Let $\widehat{V}_t$ be the shorthand notation for $V_{\widehat{M}_t}^{\widehat{\pi}_t}$, the optimal value function in the approximate model. We fix an (x,a) pair, then use the shorthands $p(y) = P(x,a,y)$ and $\widehat{p}_t(y) = \widehat{P}_t(x,a,y)$.

Bounding the number of updates. The $\widehat{\tau}$ for an SA-pair (x,a) can be updated once when it is added to the set K_t, and $V_{\max}/\epsilon_2$ times when the probabilities are updated. To see the latter, let $t > t_1$ be two consecutive update times (with the possibility $t_1 = 0$),

and note that the update condition can be written as

$$\sum_{y\in X} \widehat{p}_{t_1}(y)\widehat{V}_{t_1}(y) - \widehat{p}_t(y)\widehat{V}_t(y) \geq \epsilon_2 \,.$$

Therefore, the quantity $\sum_{y\in X} \widehat{p}_{t_k}(y)\widehat{V}_{t_k}(y)$, $k = 0, 1, 2, \ldots$, decreases by at least ϵ_2 on every update (maybe more, if $Q'(x,a) \leq \widehat{Q}_t(x,a)$). It is at most $V_{\max}$, and is always nonnegative, so at most $V_{\max}/\epsilon_2$ updates can happen because of each SA-pair. Thus, the total number of updates is at most $\kappa \stackrel{\text{def}}{=} NK(V_{\max}/\epsilon_2 + 1)$.

The value function is monotonous. $\widehat{V}_t$ is monotonously decreasing: if there is no update at time $t+1$, then $\widehat{V}_{t+1} = \widehat{V}_t$; and if there was an update in SA-pair (x_0, a_0), then $\widehat{V}_{t+1}(x_0) \leq \widehat{V}_t(x_0)$, so it is easy to see that $\widehat{V}_{t+1}(x) \leq \widehat{V}_t(x)$ for all x.

Nice set changes only when there is an update. Fix an SA-pair $(x,a) \in K_t$ and timestep t. Let $t_1 = \tau_{t,x,a} \leq t$ be the phase end preceding t. If there were no updates anywhere in the MDP between t_1 and t and $(x,a) \in L_{t_1}$, then $(x,a) \in L_t$ as well, because the set L_t can only change if there is an update to the model.

If an SA-pair is not updated twice in a row, it is "nice". Fix an SA-pair (x,a) and a time step t. With a slight abuse of notation, we denote the kth phase end of (x,a) by τ_k. For some k, $\tau_{t,x,a} = \tau_k$. Let $t_2 = \tau_{k+1}$ the random time of the phase end following t and $t_3 = \tau_{k+2}$ be the next phase end. Note that both τ_{k+1} and τ_{k+2} are potentially infinity with some probability. Let Z be the event that t_3 is finite and no update is done either at t_2 or t_3. (When we will need to refer to Z, t_2, or t_3 and the identity of t, x and a will be important, we will use the respective symbols $Z_{t,x,a}$, $t_2^{x,a}(t)$ and $t_3^{x,a}(t)$.)

For every $k \in \mathbb{N}$ and $(x,a) \in X \times A$, let $F_{k,x,a}$ be the failure event when $\sum(p(y) - \widehat{p}^{\infty}_{\tau_{k+2}}(y))\widehat{V}_{\tau_{k+1}}(y) \leq \epsilon_2$ does not hold (where $\widehat{p}^{\infty}$ is the semi-empirical approximation defined in Lemma 2). The value function $\widehat{V}_{\tau_{k+1}}$ is $\mathcal{F}_{\tau_{k+1}}$-measurable, so we can apply Lemma 2 to get $\mathbb{P}(F_{k,x,a}) \leq \delta_1$ if $m \geq \frac{8V_{\max}^2}{\epsilon_2^2} \log \frac{2}{\delta_1}$.[7] Note that the failure probability is small for each k, but their sum for all $k \in \mathbb{N}$ would still grow indefinitely. Luckily, we only have to re-check the accuracy of the approximation when the nice set changes, and by the previous point, this only happens if $\widehat{Q}_{\tau_k} \neq \widehat{Q}_{\tau_{k+1}}$. So let us define the failure event

$$F \stackrel{\text{def}}{=} \bigcup_{(x,a)\in X\times A} \bigcup_{k\in\mathbb{N}} F_{k,x,a} \cap \{\widehat{Q}_{\tau_k} \neq \widehat{Q}_{\tau_{k+1}}\} \,.$$

[7] Note that if k_l is a stopping time, we still have $\mathbb{P}(F_{k_l,x,a}) \leq \delta_1$.

Let us restrict our attention now to the event $Z \cap F^c$ (intuitively, this is the event when there are at least two more phase ends after t, we do not update the value function at either of them, and the checking of $\sum(p(y) - \widehat{p}^{\infty}_{\tau_{k+2}}(y))\widehat{V}_{\tau_{k+1}}(y) \leq \epsilon_2$ never fails). On $Z \cap F^c$, t_2 and t_3 are finite, so $\widehat{p}^{\infty}_{t_2} = \widehat{p}_{t_2}$ and $\widehat{p}^{\infty}_{t_3} = \widehat{p}_{t_3}$. Furthermore, both update checks returned with the result that an update is not necessary, so

$$\begin{aligned}
&\textstyle\sum_{y\in X} \widehat{p}_{t_1}(y)\widehat{V}_{t_1}(y) - \widehat{p}_{t_2}(y)\widehat{V}_{t_2}(y) \leq \epsilon_2 \quad \text{and} \\
&\textstyle\sum_{y\in X} \widehat{p}_{t_2}(y)\widehat{V}_{t_2}(y) - \widehat{p}_{t_3}(y)\widehat{V}_{t_3}(y) \leq \epsilon_2 \,.
\end{aligned}$$

Note that $\widehat{p}_t$ can only be updated at phase ends, so $\widehat{p}_t = \widehat{p}_{t_1}$. Using these, and the monotonicity of $\widehat{V}_t$,

$$\begin{aligned}
\sum \widehat{p}_t(y)\widehat{V}_t(y) &= \sum \widehat{p}_{t_1}(y)\widehat{V}_t(y) \leq \sum \widehat{p}_{t_1}(y)\widehat{V}_{t_1}(y) \\
&\leq \sum \widehat{p}_{t_3}(y)\widehat{V}_{t_3}(y) + 2\epsilon_2 \,; \\
\sum p(y)\widehat{V}_t(y) &\geq \sum p(y)\widehat{V}_{t_2}(y) \geq \sum \widehat{p}_{t_3}(y)\widehat{V}_{t_2}(y) - \epsilon_2 \\
&\geq \sum \widehat{p}_{t_3}(y)\widehat{V}_{t_3}(y) - \epsilon_2 \,,
\end{aligned}$$

so, on $F^c \cap Z$, $(x,a) \in L_t$ because

$$\sum (\widehat{p}_t(y) - p(y))\widehat{V}_t(y) \leq 3\epsilon_2 \,. \tag{1}$$

Bounding the failure probability. For any fixed (x,a), let $\widetilde{\kappa} = \widetilde{\kappa}(x,a)$ be the number of times when the model was updated during the interval $[\tau_k, \tau_{k+1})$. Denote the corresponding indices of τ_k by $k_1, \ldots, k_{\widetilde{\kappa}}$. The quantity $\widetilde{\kappa}$ is random, but is upper bounded by the total number of updates to the model, which is upper bounded by the deterministic quantity κ. For $\widetilde{\kappa} \leq \ell$, define $k_\ell = k_{\widetilde{\kappa}}$, then

$$\begin{aligned}
\mathbb{P}(F) &\leq \mathbb{P}\Big(\textstyle\bigcup_{(x,a)\in X\times A} \bigcup_{\ell\in\{1\ldots\kappa\}} F_{k_\ell,x,a}\Big) \\
&\leq \textstyle\sum_{x,a\in X\times A} \sum_{\ell=1}^{\kappa} \mathbb{P}(F_{k_\ell,x,a}) \leq \kappa NK\delta_1 \,.
\end{aligned}$$

Bound on the number of times when encountering a non-nice SA-pair. We wish to bound the cardinality of $B = \{t | (x_t, a_t) \notin L_t\}$ on F^c. Since we have shown that on $F^c \cap Z_{t,x,a}$, $(x,a) \in L_t$ holds, it follows that, $B \subset \{t \,|\, \omega \in Z^c_{t,x_t,a_t}\}$ (on F^c). Now, $Z^c_{t,x,a}$ happens when either $t_3^{x,a}(t)$ is infinite *or* an update is done at $t_2^{x,a}(t)$ or $t_3^{x,a}(t)$. Therefore, on F^c, $B \subset \cup_{(x,a)\in X\times A}\{t \,|\, (x,a) = (x_t,a_t), t_3^{x_t,a_t}(t) = \infty\} \cup \{t \,|\, \text{update at } t_2^{x_t,a_t}(t)\} \cup \{t \,|\, \text{update at } t_3^{x_t,a_t}(t)\}$. The cardinality of the first set in the right-hand side is at most $2mNK$, since for each SA-pair (x,a) which is visited a finite number of times, there is a time when it is visited the last time. Then, $t_3^{x,a}(t) = \infty$ can only happen at most $2m$ previous visits of (x,a) before this last visit. In order to bound the cardinality of the second set, recall that the total number of updates is κ.

Pick any update with update time u. If $u = t_2^{x_t,a_t}(t)$ for some time t, then t must be one of the at most m visits to (x_t, a_t) that happened just before u. Since this way all elements of the said set are covered, the set's cardinality cannot be larger than $m\kappa$. The same reasoning holds for the third set, showing that the total cardinality is bounded by $2m\kappa + 2mNK$.

Counting non-optimal steps. Let $T_{\text{escape}} \stackrel{\text{def}}{=} \sum_{t=0}^{\infty} \mathbb{I}\{e_t > \epsilon_1/V_{\max}\}$ be the number of time steps where the chance of running into a non-nice SA-pair in H steps is high.

Lemma 3 *Let F' be the failure event that $T_{escape} > 12mNKH^2 + 2H\sqrt{12\frac{\epsilon_1}{V_{\max}}mNKH\log\frac{1}{\delta_1}} + 4H\log\frac{1}{\delta_1}$. Then $\mathbb{P}(F') \le \delta_1$.*

Next, we will show that except for these at most T_{escape} time steps, our algorithm is near-optimal.

3.4. Proving near-optimality

Fix a time step t and let us assume that $e_t \le \epsilon_1/V_{\max}$.

Truncation. All rewards are nonnegative, so horizon-H truncation can only decrease values, so

$$V_{t,M}^{\text{MORMAX}}(x) \ge V_{t,M}^{\text{MORMAX}}(x;H) \text{ for all } x \in X: \quad (2)$$

Substitution with a stationary policy and an auxiliary model. The event that the model is updated in the next H steps is contained in the event E_t which has low probability: $e_t = \mathbb{P}(E_t|\mathcal{F}_t) \le \epsilon_1/V_{\max}$. Furthermore, the event that the agent leaves L_t in the next H steps is also contained in E_t. Therefore, on $(E_t)^c$, the policy of MORMAX stays stationary at $\widehat{\pi}_t$, and it does not leave the set L_t. We also know that on L_t, M and $\overline{M}_t$ are identical. Therefore,

$$V_M^{\text{MORMAX}}(x_t;H) \ge V_{\overline{M}_t}^{\widehat{\pi}_t}(x_t;H) - \epsilon_1\,. \quad (3)$$

Undo truncation. If the horizon time $H \ge \frac{1}{1-\gamma}\log\frac{V_{\max}}{\epsilon_1}$, the effect of truncation is small enough to give

$$V_{\overline{M}_t}^{\widehat{\pi}_t}(x;H) \ge V_{\overline{M}_t}^{\widehat{\pi}_t}(x) - \epsilon_1 \text{ for all } x \in X. \quad (4)$$

Approximation with the empirical model. If the transition probabilities of two MDPs are close to each other, then so are the value functions of a fixed policy:

Lemma 4 *Let $M_1 = (X, A, P_1, R)$, $M_2 = (X, A, P_2, R)$ be two MDPs, and fix discount factor γ, precision ϵ_1, and policy π. Let V_1 and V_2 be the value functions of π in their respective MDPs. If*

$$\textstyle\sum_{y\in X}(P_1 - P_2)(x,a,y)V_1(y) \le \epsilon_1(1-\gamma)/\gamma$$

for every $(x,a) \in X \times A$, then $V_1(x) - V_2(x) \le \epsilon_1$ for every $x \in X$.

For any $(x,a) \in L_t$, $\overline{P}_t(x,a,y) = P(x,a,y)$. Recall that L_t is the set of SA-pairs where the effect of changing the model is in some sense small: $L_t = \{(x,a) \in K_t : \sum_{y\in X}(\widehat{P}_t - P)(x,a,y)V_{\widehat{M}_t}^{\widehat{\pi}_t}(y) \le 3\epsilon_2\}$. On $(L_t)^c$, $\overline{P}_t(x,a,y) = \widehat{P}(x,a,y)$, so $\sum_{y\in X}(\widehat{P}_t - \overline{P}_t)(x,a,y)V_{\widehat{M}_t}^{\widehat{\pi}_t}(y) \le 3\epsilon_2 = \epsilon_1(1-\gamma)/\gamma$ holds for every SA-pair, and by Lemma 4 this means that

$$V_{\overline{M}_t}^{\widehat{\pi}_t}(x) \ge V_{\widehat{M}_t}^{\widehat{\pi}_t}(x) - \epsilon_1 \text{ for all } x \in X. \quad (5)$$

Optimality of $\widehat{\pi}_t$. The optimal policy of $\widehat{M}_t$ is $\widehat{\pi}_t$, so its value is at least as large as for any other policy, for example π^*:

$$V_{\widehat{M}_t}^{\widehat{\pi}_t}(x) \ge V_{\widehat{M}_t}^{\pi^*}(x) \text{ for all } x \in X. \quad (6)$$

Substitution with another auxiliary model. Let us introduce now a new auxiliary MDP $\widetilde{M}_t = (X, A, \widetilde{P}_t, R)$ that is identical to $\widehat{M}_t$ on K_t and to M outside: $\widetilde{P}_t(x,a,y) \stackrel{\text{def}}{=} \begin{cases} \widehat{P}(x,a,y), & \text{if } (x,a) \in K_t; \\ P(x,a,y), & \text{if } (x,a) \notin K_t. \end{cases}$

The following lemma tells that if the one-step look-aheads are consistently smaller in one MDP than in an other, then the value function of a fixed policy is also smaller there:

Lemma 5 *Consider two MDPs $M_1 = (X, A, P_1, R_1)$, $M_2 = (X, A, P_2, R_2)$, and fix discount factor γ and stationary policy π. Let V_1 and V_2 be the value functions of π in their respective MDPs and let $v \in \mathbb{R}^X$ be an arbitrary value function. If*

$$R_1(x,a) + \gamma\sum_{y\in X}P_1(x,a,y)v(y) \le R_2(x,a) + \gamma\sum_{y\in X}P_2(x,a,y)v(y)$$

for every $(x,a) \in X \times A$, then $V_1(x) \le V_2(x)$, $\forall x \in X$.

Applying the lemma to $\widehat{M}_t$ and $\widetilde{M}_t$, we get that

$$V_{\widehat{M}_t}^{\pi^*}(x) \ge V_{\widetilde{M}_t}^{\pi^*}(x) \text{ for all } x \in X. \quad (7)$$

Substitution with the real model. For all $(x,a) \in X \times A$, consider the quantity

$$d_t(x,a) = \textstyle\sum_{y\in X}(P(x,a,y) - \widetilde{P}_t(x,a,y))V^*(y).$$

For $(x,a) \notin K_t$, $\widetilde{P}_t(x,a,y) = P(x,a,y)$, so $d_t(x,a) =$

0. For $(x,a) \in K_t$, $\widehat{P}_t = \widehat{P}_{\widehat{\tau}_k}$ for some $k > 0$ with $\widehat{\tau}_k \le t$. The optimal value function V^* is $\mathcal{F}_{\widehat{\tau}_{k-1}}$-measurable (as it is non-random), so, by Lemma 2, if $m \ge \frac{8V_{\max}^2}{\epsilon_2^2} \log \frac{2}{\delta_1}$, then $\sum_{y\in X}(P(x,a,y) - \widehat{P}^\infty(x,a,y))V^*(y)$ except for an event $F''_{x,a}$ with probability $\mathbb{P}(F''_{x,a}) \le \delta_1$. Because of $\widehat{\tau}_k \le t$, $\widehat{\tau}_k$ is finite, so $\widehat{P}^\infty(x,a,y) = \widehat{P}_t(x,a,y) = \widetilde{P}_t(x,a,y)$ for all $y \in X$, so $d_t(x,a) \le \epsilon_2 < 3\epsilon_2 = \epsilon_1(1-\gamma)/\gamma$.

Let $F'' = \bigcup_{x,a\in X\times A} F''_{x,a}$, for which $\mathbb{P}(F'') \le NK\delta_1$. Consider now the case when event $(F'')^c$ holds. In this case, by Lemma 4,

$$V^{\pi^*}_{\widetilde{M}_t}(x) \ge V^{\pi^*}_M(x) - \epsilon_1 \text{ for all } x \in X. \tag{8}$$

Bounding failure. The algorithm may encounter three failure events: F, F' and F''. Their total probability is $\mathbb{P}(F \cup F' \cup F'') \le \mathbb{P}(F) + \mathbb{P}(F') + \mathbb{P}(F'') \le N^2 \cdot K^2 \cdot (V_{\max}/\epsilon_2 + 1)\delta_1 + \delta_1 + NK\delta_1$ This is at most δ for $\delta_1 = \delta/(4N^2K^2V_{\max}/\epsilon_2)$. Apart from this less-than-δ chance of failure, the algorithm will make at most $12mNKH^2 + 2H\sqrt{12\frac{\epsilon_1}{V_{\max}}mNKH\log\frac{1}{\delta_1}} + 4H\log\frac{1}{\delta_1}$ non-near-optimal steps. In the rest of the time, by adding up eqs. (2)–(8), $V_M^{\textsc{Mormax}}(x_t) \ge V_M^*(x) - 5\epsilon_1$. If $\epsilon_1 = \epsilon/5$, we get ϵ-optimality. Expressed with the original parameters, the necessary sample size is

$$m = \frac{8V_{\max}^2}{\epsilon_2^2}\log\frac{2}{\delta_1} \le \frac{1800\gamma^2V_{\max}^2}{\epsilon^2(1-\gamma)^2}\log\frac{144\gamma N^2K^2V_{\max}}{\delta\epsilon(1-\gamma)}$$

and the upper bound on the non-optimal time steps is $\widetilde{O}\left(\frac{NKV_{\max}^2}{(1-\gamma)^4\epsilon^2}\right)$.

4. Discussion

For the sake of simplicity, we assumed that the reward function is known, and only the transitions have to be learned. When the reward is also learnt, we need to make only slight modifications to the algorithm and the analysis. For example, instead of bounding $\sum_y(\widehat{P}_t(x,a,y) - P(x,a,y))V(y)$, we need to bound $\sum_y(\widehat{R}(x,a) + \gamma\widehat{P}_t(x,a,y)V(y)) - (R(x,a) + \gamma P(x,a,y)V(y))$. The sample complexity bounds remain exactly the same.

Of all PAC-MDP algorithms, proving the bounds of Rmax is the simplest. Nevertheless, the efficiency of the other listed methods is proven using the same tricks, so we believe that their complexity upper bounds can be improved likewise, so that the dependence on N is log-linear.

A strange property of Mormax is that when a new batch of samples is collected in an SA-pair, it throws away the old samples. The flow of fresh samples is crucial for the proof. However, from a practical point of view, throwing away information seems suboptimal, and is not clear whether it is really necessary. We also note that sample complexity bounds, including ours, are extremely conservative, because they are for the worst case. In practice, much lower values of m are used, so re-estimation of the probabilities can have a significant effect. On the other hand, preserving old samples could have more benefits, too.

Acknowledgments

This research was supported by iCORE and Alberta Ingenuity, both part of Alberta Innovates –Technology Futures, NSERC and the PASCAL2 Network of Excellence under EC grant no. 216886. Cs. Szepesvári is on leave from MTA SZTAKI.

References

Auer, P., Jaksch, T., and Ortner, R. Near-optimal regret bounds for reinforcement learning. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L. (eds.), *NIPS-21*, pp. 89–96. MIT Press, 2009.

Bertsekas, D. P. and Tsitsiklis, J. N. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA, 1996.

Brafman, R. I. and Tennenholtz, M. R-MAX - a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3:213–231, 2002.

Kakade, S.M. *On the sample complexity of reinforcement learning*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London, 2003.

Kearns, M. and Singh, S. Finite-sample convergence rates for Q-learning and indirect algorithms. In Jordan, M. I., Kearns, M. J., and Solla, S. A. (eds.), *NIPS-10*, pp. 996–1002. MIT Press, 1998.

Kearns, M.J. and Singh, S.P. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2–3): 209–232, 2002.

Li, L. *A Unifying Framework for Computational Reinforcement Learning Theory*. PhD thesis, Rutgers University, New Brunswick, NJ, USA, October 2009.

Strehl, A. L. and Littman, M. L. A theoretical analysis of model-based interval estimation. In De Raedt, L. and Wrobel, S. (eds.), *ICML 2005*, pp. 857–864. ACM, 2005.

Strehl, A. L., Li, L., Wiewiora, E., Langford, J., and Littman, M. L. PAC model-free reinforcement learning. In Cohen, W. W. and Moore, A. (eds.), *ICML 2006*, pp. 881–888. ACM, 2006.

Szita, I. and Lőrincz, A. The many faces of optimism: a unifying approach. In Cohen, W. W., McCallum, A., and Roweis, S. T. (eds.), *ICML 2008*, pp. 1048–1055. ACM, 2008.