A Theoretical Analysis of Model-Based Interval Estimation

Alexander L. Strehl STREHL@CS.RUTGERS.EDU
Michael L. Littman MLITTMAN@CS.RUTGERS.EDU
Department of Computer Science, Rutgers University, Piscataway, NJ USA

Abstract

Several algorithms for learning near-optimal policies in Markov Decision Processes have been analyzed and proven efficient. Empirical results have suggested that Model-based Interval Estimation (MBIE) learns efficiently in practice, effectively balancing exploration and exploitation. This paper presents the first theoretical analysis of MBIE, proving its efficiency even under worst-case conditions. The paper also introduces a new performance metric, average loss, and relates it to its less "online" cousins from the literature.

1. Introduction

In the reinforcement-learning problem, agents learn by experimentation to maximize a performance objective. The underlying mathematical framework generally used is that of Markov Decision Processes or MDPs (Puterman, 1994). This paper considers discounted infinite-horizon MDPs with stochastic reward and transition functions.

While there are many learning algorithms for MDPs, only a few have been shown to produce near-optimal policies, with high probability, after a polynomial amount of experience. Such algorithms are said to be *probably approximately correct* (PAC). The E^3 algorithm (Kearns & Singh, 2002) and the conceptually simpler Rmax (Brafman & Tennenholtz, 2002) are two state-of-the-art PAC reinforcement-learning algorithms. They have similar worst-case bounds.

Model-based Interval Estimation (MBIE) is another learning algorithm that builds a model to construct an exploration policy (Wiering & Schmidhuber, 1998; Strehl & Littman, 2004). In contrast to Rmax and E^3, MBIE incorporates acquired experience more quickly and smoothly into its internal model. However, the rate at which MBIE learns had not been analyzed in the PAC framework. This paper first presents an overview of the different definitions of efficient learning that have been used in analyses. Another, more "online", and therefore more realistic, definition is introduced and related to the sample complexity notion of Kakade (2003). Then, a PAC analysis of MBIE's sample complexity is performed, producing a worst-case result comparable to that of Rmax, complementing MBIE's strong empirical performance.

2. Notation

This section introduces the Markov Decision Process (MDP) notation used throughout the paper; see Sutton and Barto (1998) for an introduction. An MDP M is a five tuple $\langle S, A, T, R, \gamma \rangle$, where S is the state space, A is the action space, $T : S \times A \times S \to \mathbb{R}$ is a transition function, $R : S \times A \to \mathbb{R}$ is a reward function, and $0 \leq \gamma < 1$ is a discount factor on the summed sequence of rewards. From state s under action a, the agent receives a sample reward from a distribution with mean $R(s, a)$ and is transported to state s' with probability $T(s, a, s')$. We assume that there are a finite number of possible rewards, all of which lie between 0 and a positive real number $R_{\max}$.[1] For a stationary policy π, let $V^\pi(s)$ ($Q^\pi(s, a)$) denote the value (action-value) function for π in M (which may be omitted from the notation) from state s. For simplicity, all policies in this paper are assumed to be deterministic (it's possible to extend our results to stochastic policies). The optimal policy is denoted π^* and has value functions $V^*_M(s)$ and $Q^*_M(s, a)$. Note that a policy cannot have a value greater than $v_{\max} := \frac{R_{\max}}{1-\gamma}$. If T is a positive integer, let $V^\pi_M(s, T)$ denote the T-step value function of policy π. If π is non-stationary, then s is replaced by a *partial path* $c_t = s_1, a_1, r_1, \ldots, s_t$, in the previous definitions. Specifically, let s_t and r_t be the tth encountered state and received reward,

Appearing in *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, Germany, 2005.

[1] This finite-reward assumption generalizes the notion of deterministic rewards used in prior analytic work.

respectively, resulting from execution of policy π in some MDP M. Then, $V_M^\pi(c_t) = E[\sum_{j=0}^{\infty} \gamma^j r_{t+j}]$ and $V_M^\pi(c_t, T) = E[\sum_{j=0}^{T-1} \gamma^j r_{t+j}]$. These expectations are with respect to some fixed prefix sequence c_t and are taken over all possible infinite paths the agent might follow from the tth step and onward. Note that we may omit the actions a_i and refer to c_t as a *partial sequence* where $c_t = s_1, r_1, \ldots, s_t$. In our analysis, we only consider *deterministic* learning algorithms and, thus, these two definitions are equivalent.

3. Performance Metrics

A reasonable notion of learning efficiency in an MDP is to require an efficient algorithm to achieve near-optimal (expected) performance with high probability. An algorithm that satisfies such a condition can be said to be *probably approximately correct* or PAC. The PAC notion was developed in the supervised learning community, where a classifier, while learning, does not influence the distribution of training instances it receives. In reinforcement learning, learning and behaving are intertwined, with the decisions made during learning profoundly affecting the available experience.

In applying the PAC notion in the reinforcement-learning setting, researchers have examined definitions that vary in the degree to which the natural mixing of learning and evaluation is restricted for the sake of analytic tractability. We survey these notions next.

Fiechter (1997) explored a set of PAC-learning definitions that assumed that learning is conducted in trials of constant length from a fixed start state. Under this *reset assumption*, the task of the learner is to find a near-optimal policy from the start state given repeated visits to this state.

Kearns and Singh (2002) observed that the reset assumption is not strictly necessary. In any sufficiently long run, there must be some state that is repeatedly visited and can therefore serve as a kind of *post hoc* starting state for analysis. They showed that a PAC result could be derived for trajectory-based learning instead of assuming independent trials.

In this setting, a learning algorithm is judged by whether it is guaranteed to reach a state, after a polynomial number of steps, for which it can output an ϵ-optimal policy (from that state) with probability at least $1 - \delta$. Kearns and Singh (2002) provided an algorithm they called E^3 that satisfies this PAC notion.

Kakade (2003) introduced a PAC performance metric that is more "online" in that it evaluates the behavior of the learning algorithm itself as opposed to a separate policy that it outputs. As in Kearns and Singh's definition, learning takes place over one long path through the MDP. At time t, the partial path $c_t = s_1, a_1, r_1 \ldots, s_t$ is used to determine a next action a_t. The algorithm itself can be viewed as a non-stationary policy. In our notation, this policy has expected value $V^{\mathcal{A}}(c_t)$, where $\mathcal{A}$ is the learning algorithm.

Definition 1 *(Kakade, 2003) The* **sample complexity of exploration** *of an algorithm $\mathcal{A}$ is the number of timesteps t such that $V^{\mathcal{A}}(c_t) < V^*(s_t) - \epsilon$.*

In other words, the sample complexity is the number of timesteps, over the course of any run, for which the learning algorithm $\mathcal{A}$ is not executing an ϵ-optimal policy from its current state. $\mathcal{A}$ is PAC in this setting if its sample complexity can be bounded by a number polynomial in the relevant quantities with high probability. Kakade showed that the Rmax algorithm (Brafman & Tennenholtz, 2002) satisfies this condition.

Although sample complexity demands a tight integration between behavior and evaluation, the evaluation itself is still in terms of the near-optimality of expected values over future policies as opposed to the actual rewards the algorithm achieves while running. We introduce a new performance metric, *average loss*, defined in terms of the actual rewards received by the algorithm while learning. In the remainder of the section, we define average loss formally. In the next section, we show that efficiency in the sample-complexity setting implies efficiency in the average-loss setting.

Definition 2 *Suppose a learning algorithm is run for one trial of T steps in an MDP M. Let s_t be the state encountered on step t and let r_t be the tth reward received. Then, the* **instantaneous loss** *of the agent is $il(t) = V^*(s_t) - \sum_{i=t}^{T} \gamma^{i-t} r_i$, the difference between the optimal value function at state s_t and the actual discounted return of the agent from time t until the end of the trial. The quantity $l = \frac{1}{T} \sum_{t=1}^{T} il(t)$ is called the* **average loss** *over the sequence of states encountered.*

In this setting, a learning algorithm is PAC if, for any ϵ and δ, we can choose a value T, polynomial in the relevant quantities $(1/\epsilon, 1/\delta, |S|, |A|, 1/(1-\gamma), R_{\max})$, such that the average loss of the agent (following the learning algorithm) on a trial of T steps is guaranteed to be less than ϵ with probability at least $1 - \delta$.

4. Average Loss & Sample Complexity

This section shows that a PAC algorithm in the sample-complexity framework is also PAC under av-

erage loss. Since average loss is arguably a more natural performance metric for learning in MDPs, while sample complexity admits a cleaner and more direct analysis, this result provides the best of both worlds.

4.1. Properties of Adjusted Average Loss

Two properties of Definition 2 present some bookkeeping difficulties. First, instantaneous loss compares the expected return of the optimal policy over an infinite length sequence ($V^*(s_t)$) to the return of the learning algorithm over a finite length path T. Second, the length of the finite sequence is variable and depends on the current time. The complexity of these properties is mitigated by the following definitions.

Definition 3 *Suppose a learning algorithm is run for one sequence of $T_1 + T_2 - 1$ steps. Let c_t be the partial sequence $s_1, r_1, \ldots, s_{t-1}, r_{t-1}, s_t$. For any policy π and integer t such that $t \leq T_1$, let $R^\pi_{T_2}(t) := \sum_{t'=t}^{t+T_2-1} \gamma^{t'-t} r_{t'} + \gamma^{T_2} V^\pi(c_{t+T_2})$ be the* **adjusted return**. *Let $I^\pi_{T_2}(t) := V^\pi(c_t) - R^\pi_{T_2}(t)$ be the* **adjusted instantaneous loss**. *Let $L^\pi_{T_1,T_2} = \frac{1}{T_1} \sum_{t=1}^{T_1} I^\pi_{T_2}(t)$ be the* **adjusted average loss**.

Adjusted return is the actual discounted sum of rewards starting at step t over the next T_2 steps, plus the discounted expected return π would receive starting from the state reached T_2 steps in the future. Adjusted instantaneous loss is the true return for policy π from time t minus the adjusted return—how much was lost relative to simply following π. Adjusted average loss is the average of the adjusted instantaneous losses over the first T_1 steps of the run. In these definitions, the policy π is not required to be the same as the policy followed by the algorithm.

For any (possibly nonstationary) policy π, MDP M, and integer T, we can run π in M for T steps. Let the partial sequence c_T be the list of states and rewards encountered by the agent along this run. Each time this experiment is performed, a different sequence might be generated. Thus, we say that c_T is *to be generated by* π, to emphasize the fact that c_T is a random partial sequence. Note that the adjusted instantaneous loss and adjusted average loss quantities are random variables dependent on the relevant partial sequence. We will find it useful to define the following additional random variables, $Y^\pi_t := V^\pi(c_t) - (r_t + \gamma V^\pi(c_{t+1}))$, for all $t < T$. As usual, in this definition, c_t is the partial sequence consisting of the prefix of c_T ending at state s_t (the tth state encountered). It follows from our definition that as long as the agent follows π, the expectation of Y^π_t is zero—it is the Bellman error in the value-function update for π.

Consider the sequence $Z := Y^\pi_1, Y^\pi_2 \ldots, Y^\pi_T$ of random variables up to time T. Next, we will show that any subsequence q of Z is a *martingale difference sequence*, meaning that the each term in q has expectation zero even when conditioned on all previous terms of q.

Lemma 1 *Let π be a policy, and suppose the sequence $s_1, r_1, s_2, r_2, \ldots, s_T, r_T$ is to be generated by π. If $1 \leq q_1 < q_2 < \cdots < q_i < t \leq T$, then $E[Y^\pi_t | Y^\pi_{q_1}, Y^\pi_{q_2} \ldots, Y^\pi_{q_i}] = 0$.*

Proof: Let $[Y^\pi_t | c_{t+1}]$ be the value of the random variable Y^π_t given the fixed partial sequence c_{t+1}. Then,

$$\begin{aligned} E[Y^\pi_t] &= \sum_{c_{t+1}} \Pr(c_{t+1})[Y^\pi_t | c_{t+1}] \\ &= \sum_{c_t} \Pr(c_t) \sum_{r_t, s_{t+1}} \Pr(r_t, s_{t+1} | c_t)[Y^\pi_t | c_t, r_t, s_{t+1}]. \end{aligned}$$

The sum in the first line above is over all possible sequences $c_{t+1} = s_1, r_1, \ldots, s_{t+1}$ resulting from t action choices by an agent following policy π.

In the term above, we note that conditioning Y^π_t on the sequence of random variables $Y^\pi_{q_1}, Y^\pi_{q_2} \ldots, Y^\pi_{q_i}$ can certainly affect the probabilities $\Pr(c_t)$, by making some sequences more likely and others less likely. However, the term $\sum_{c_t} \Pr(c_t)$ will always be one. Notice that fixed values of $Y^\pi_{q_1}, Y^\pi_{q_2} \ldots, Y^\pi_{q_i}$ cannot influence the innermost sum. Now, we have that

$$\begin{aligned} &\sum_{r_t, s_{t+1}} \Pr(r_t, s_{t+1} | c_t)[Y^\pi_t | c_t, r_t, s_{t+1}] \\ &= V^\pi(c_t) - \sum_{r_t, s_{t+1}} \Pr(r_t, s_{t+1} | c_t)(r_t + \gamma V^\pi(c_{t+1})). \end{aligned}$$

By the definition of $V^\pi(c_t)$, this last term is zero. □

Adjusted instantaneous loss can now be reformulated as the discounted sum of these random variables.

Lemma 2 *If $t \leq T_1$ is a positive integer, then $I^\pi_{T_2}(t) = \sum_{t'=t}^{t+T_2-1} \gamma^{t'-t} Y^\pi_{t'}$.*

4.2. Adjusted and Average Loss

This section shows that the quantities T_1 and T_2, the number and length of the trials in Definition 3, may be only polynomially large and still ensure that results about adjusted loss apply to average loss.

Proposition 1 *Suppose that $l \geq 0$. If $T_1 \geq \frac{2T_2 R_{\max}}{\epsilon}$ and $T_2 \geq \ln(\frac{\epsilon(1-\gamma)}{2R_{max}})/\ln(\gamma)$, then $l - L^{\pi^*}_{T_1,T_2} \leq \epsilon$.*

The importance of the result, proven elsewhere (Strehl & Littman, 2005), is that we can bound the average loss l by bounding the adjusted loss $L^\pi_{T_1,T_2}$.

4.3. Reduction to Sample Complexity

Our main objective here is to relate sample complexity and average loss. We now show that the number of trials T_1 used in the adjusted definition of average loss can be made large enough (but not more than polynomially large) so that, with high probability, any algorithm's average loss can be made arbitrarily small given that the algorithm's sample complexity is bounded with high probability.

Proposition 2 *Suppose T_2 and C are two positive integers. If C is a bound on the sample complexity of some algorithm $\mathcal{A}$ with respect to ϵ, which holds with probability at least $1-\delta$, then T_1 can be chosen so that the adjusted average loss $L^{\pi^*}_{T_1,T_2} \leq 3\epsilon$, with probability at least $1-2\delta$.*

Proof: We consider running algorithm $\mathcal{A}$, which can be viewed as a non-stationary policy, in the MDP for $T := T_1 + T_2 - 1$ steps. Partition the generated partial sequence $s_1, r_1, \ldots, s_{T_1}, r_{T_1}$, into those timesteps $t \in S_B$ such that $\mathcal{A}$ is not ϵ-optimal, and those timesteps $t \in S_G$ such that it is. Now,

$$L^{\pi^*}_{T_1,T_2} = \frac{1}{T_1}\sum_{t=1}^{T_1} I^{\pi^*}_{T_2}(t) = \frac{1}{T_1}\sum_{t\in S_G} I^{\pi^*}_{T_2}(t) + \frac{1}{T_1}\sum_{t\in S_B} I^{\pi^*}_{T_2}(t).$$

By the sample-complexity bound, with high probability, $|S_B| \leq C$. Combining this fact with the fact that $I^{\pi^*}_{T_2}$ can be at most $v_{\max}$ (which must be nonnegative due to our assumption of nonnegative rewards) yields:

$$L^{\pi^*}_{T_1,T_2} \leq \frac{1}{T_1}\sum_{t\in S_G} I^{\pi^*}_{T_2}(t) + \left(\frac{1}{T_1}\right)(C)(v_{\max}). \quad (1)$$

Restricting to $t, t' \in S_G$, Lemma 2 reveals that

$$\begin{aligned}\sum_t I^{\mathcal{A}}_{T_2}(t) &= \sum_t \sum_{t'=t}^{t+T_2-1} \gamma^{t'-t} Y^{\mathcal{A}}_{t'} \\ &= \sum_{t'=1}^{T_2-1}\sum_{t=1}^{t'} \gamma^{t'-t} Y^{\mathcal{A}}_{t'} + \sum_{t'=T_2}^{T_1+T_2+1}\sum_{t=t'-T_2+1}^{T_1} \gamma^{t'-t} Y^{\mathcal{A}}_{t'} \\ &\leq \sum_{t'=1}^{T_2-1} Y^{\mathcal{A}}_{t'}\frac{\gamma^{t'}-1}{\gamma-1} + \sum_{t'=T_2}^{T_1+T_2-1} Y^{\mathcal{A}}_{t'}\frac{\gamma^{T_2}-\gamma^{t'-T_1}}{\gamma-1}.\end{aligned}$$

The second line above results from switching the order of the summands, which allows us to evaluate the innermost sums of that line. The last line reveals that $\sum_{t\in S_G} I^{\mathcal{A}}_{T_2}(t)$ is the sum of a martingale difference sequence, where each term is bounded by $v_{\max}/(1-\gamma)$. Therefore, applying Azuma's Lemma (Strehl & Littman, 2005) yields

$$P\left(\sum_{t\in S_G} I^{\mathcal{A}}_{T_2}(t) > a\right) \leq \exp\left(\frac{-a^2(1-\gamma)^2}{2{v_{\max}}^2(T_1+T_2-1)}\right). \quad (2)$$

For all $t \in S_G$, $I^{\pi^*}_{T_2}(t) - I^{\mathcal{A}}_{T_2}(t) \leq \epsilon$ holds since $\mathcal{A}(c_t)$ is ϵ-optimal. By Equation 2, $\frac{1}{T_1}\sum_{t\in S_G} I^{\pi^*}_{T_2}(t) \leq 2\epsilon$ with high probability when $\exp\left(\frac{-T_1^2\epsilon^2(1-\gamma)^2}{2{v_{\max}}^2(T_1+T_2-1)}\right) \leq \delta$. This condition is equivalent to the following:

$$\begin{aligned}&T_1(T_1\epsilon^2(1-\gamma)^2 - 2\ln(1/\delta){v_{\max}}^2) \\ &\geq \quad 2\ln(1/\delta){v_{\max}}^2(T_2-1). \quad (3)\end{aligned}$$

Equation 3 is satisfied when the following holds:

$$T_1 \geq \max\{\frac{1+2\ln(1/\delta){v_{\max}}^2}{\epsilon^2(1-\gamma)^2}, 2\ln(1/\delta){v_{\max}}^2(T_2-1)\}. \quad (4)$$

Finally, to ensure that the second term of Equation 1 is no more than ϵ, it is sufficient to enforce the following inequality:

$$T_1 \geq \frac{1}{\epsilon}(C)(v_{\max}). \quad (5)$$

Note that T_1 can satisfy Equations 4 and 5, yet still be no more than polynomial in the relevant quantities $1/\delta$, $1/\epsilon$, $v_{\max}$, $1/(1-\gamma)$, C, and T_2. □

In summary, low sample complexity implies low average loss since the algorithm does not have to run too long before the number of near-optimal trials is sufficient to make the average loss low.

5. Model-Based Interval Estimation

The core idea of MBIE was first introduced by Wiering and Schmidhuber (1998); however the form of the confidence intervals were *ad hoc* and problematic for analysis. We will analyze a more statistically justified approach due to Strehl and Littman (2004). This section provides a detailed description of the inner workings of MBIE, while Section 7 provides the first proof that MBIE is PAC in the sample-complexity framework. From Proposition 2, this result implies that MBIE is also PAC by the average-loss metric.

5.1. Description of the MBIE Algorithm

MBIE is a generalization of the Interval Estimation (IE) algorithm for the k-armed bandit problem (Kaelbling, 1993). MBIE, like IE, works by constructing confidence intervals on possible models based on experience. It then assumes that the most optimistic model

consistent with the data is true and behaves optimally according to this model.[2] If the agent's model is accurate, near-optimal reward (in expectation) is achieved. Otherwise, new experience is obtained and used to update the model.

More specifically, at each step, MBIE uses the available experience to determine an MDP $\tilde{M}$ with transition function $\tilde{T}$ and reward function $\tilde{R}$. This internal model has value function $\tilde{Q}$ and at least one optimal policy $\tilde{\pi}$, which MBIE uses to choose its next action. Of course, with only a finite amount of experience, MBIE cannot hope to model the environment with complete accuracy. MBIE quantifies its certainty by maintaining confidence intervals for each possible source of uncertainty. At any given stage during training, there are many possible MDPs consistent with MBIE's confidence intervals. Among these, MBIE chooses to act according to a model for which the agent can achieve maximum possible reward.

MBIE uses its experience very naturally in that its model is updated immediately as new experience arrives. In contrast, the PAC reinforcement-learning algorithms Rmax and E^3 only allow a fixed number of model updates, so at any given step in the algorithm much of the agent's experience is ignored.

In the following sections, we describe the precise form of the confidence intervals that MBIE maintains and how they are combined to produce an efficient reinforcement-learning algorithm.

5.2. The Reward Confidence Interval

For a fixed state-action pair (s, a), let $\hat{R}(s, a)$ be the sample mean of the observed rewards and $n(s, a)$ be the number of times action a has been chosen in state s. The reward assumed by MBIE's model is $\tilde{R}(s, a) = \hat{R}(s, a) + \epsilon^R_{n(s,a)}$, where $\epsilon^R_{n(s,a)} := \sqrt{\frac{\ln(2/\delta_R){R_{\max}}^2}{2n(s,a)}}$. This expression gives us the upper confidence interval on the mean reward by a straightforward application of the Hoeffding bound.

5.3. The Transition Confidence Interval

For a fixed state-action pair (s, a), let $T(s, a, \cdot)$ be the true transition probability vector and $\hat{T}(s, a, \cdot)$ the empirical distribution. With probability at least $1 - \delta_T$, the L_1 distance between $T(s, a, \cdot)$ and $\hat{T}(s, a, \cdot)$ will be at most

$$\epsilon^T_{n(s,a)} = \sqrt{\frac{2[\ln(2^{|S|} - 2) - \ln(\delta_T)]}{n(s, a)}}.$$

This result (Weissman et al., 2003) yields a confidence interval of

$$CI = \{\tilde{T}(s, a, \cdot) \mid ||\tilde{T}(s, a, \cdot) - \hat{T}(s, a, \cdot)||_1 \leq \epsilon^T_{n(s,a)}\}.$$

5.4. Combining the Confidence Intervals

Now, for each state-action pair, MBIE finds the probability distribution $\tilde{T}(s, a, \cdot)$ within CI that leads to the policy with the largest value. This quantity is formalized by the Bellman equations

$$\tilde{Q}(s, a) = \tilde{R}(s, a) + \max_{\tilde{T}(s,a,\cdot) \in CI} \gamma \sum_{s'} \tilde{T}(s, a, s') \max_{a'} \tilde{Q}(s', a'). \quad (6)$$

Note that this expression effectively combines the uncertainty in the rewards and transitions to provide the MDP model used by MBIE. Equation 6 can be solved efficiently using value iteration (Strehl & Littman, 2004). Once Equation 6 is solved, a greedy policy $\tilde{\pi}$ with respect to $\tilde{Q}$ is used by MBIE to choose the next action.

As the MBIE agent gathers experience, it is continuously updating and solving its model of the world according to Equation 6. Let C be any confidence interval computed by MBIE. We say that C is *consistent* if it contains the mean of the distribution that produced the samples for which C was computed from. For our following analysis, we require that all confidence intervals—reward and transition—be consistent for all state-action pairs over every time-step, with high probability. This condition cannot be guaranteed for fixed values of δ_R and δ_T, as we have to allow for a possibly infinite number of time steps. However, the problem can be fixed by allowing the confidence intervals for each state-action pair (s, a) to depend on the number of times $n(s, a)$ the state-action pair has been experienced. In fact, setting $\delta_R = \delta_T = \frac{3\delta}{2(|S|)(|A|)\pi^2 n(s,a)^2}$ is sufficient to ensure that every confidence interval computed by MBIE (on a given run) is consistent with probability at least $1 - \delta/2$ (Fong, 1995). Note that π here refers to the ratio of a circle's circumference to diameter, not a policy.

6. Basic Properties of MBIE

Several lemmas and basic properties of MBIE are now developed. First, for long enough time intervals, truncating the value function doesn't change it very much.

[2] We call this idea the *Pangloss assumption*—assume we are in the best of all possible worlds. The name comes from Dr. Pangloss, a character from *Candide* by Voltaire (1759), who proved this assumption "to admiration" in spite of being the victim of a series of highly unfortunate events.

Lemma 3 *If* $T \geq \frac{1}{1-\gamma}\ln\frac{R_{\max}}{\epsilon(1-\gamma)}$ *then* $|V^\pi(s,T) - V^\pi(s)| \leq \epsilon$ *for all policies* π *and states* s.

Proof: See Lemma 2 of Kearns and Singh (2002). □

The following lemma, whose proof is omitted, helps develop Lemma 5, a slight improvement over the "Simulation Lemma" of Kearns and Singh (2002) for the discounted case.

Lemma 4 *Let* $M_1 = \langle S, A, T_1, R_1, \gamma\rangle$ *and* $M_2 = \langle S, A, T_2, R_2, \gamma\rangle$ *be two MDPs with non-negative rewards bounded by* $R_{\max}$. *If* $|R_1(s,a) - R_2(s,a)| \leq \alpha$ *and* $||T_1(s,a,\cdot) - T_2(s,a,\cdot)||_1 \leq 2\beta$ *for all states* s *and actions* a, *then the following condition holds for all states* s, *actions* a *and stationary policies* π:

$$|Q_1^\pi(s,a) - Q_2^\pi(s,a)| \leq \frac{(1-\gamma)\alpha + \gamma\beta R_{\max}}{(1-\gamma)(1-\gamma+\beta\gamma)}.$$

Algorithms like MBIE act according to an internal transition and reward model. The following lemma shows that two MDPs with similar transition and reward functions have similar value functions. Thus, an agent need only ensure accuracy in the transitions and rewards of its model to guarantee near-optimal behavior. Using Lemma 4, we can prove the following result.

Lemma 5 *Let* M_1 *and* M_2 *be two MDPs as in Lemma 4. For any* $\epsilon > 0$ *and stationary policy* π, *there is a constant* C *such that if* $\alpha = 2\beta = C\left(\frac{(1-\gamma)^2\epsilon}{R_{\max}}\right)$, *then*

$$|Q_1^\pi(s,a) - Q_2^\pi(s,a)| \leq \epsilon. \quad (7)$$

The next lemma quantifies the amount of experience, for each state-action pair, required by MBIE to accurately model the dynamics of the environment.

Lemma 6 *Let* (s,a) *be a fixed state-action pair and suppose that all confidence intervals computed by MBIE are consistent. Then, there exists a positive integer* $b(\epsilon)$, *polynomial in the relevant quantities, such that* $||\tilde{T}(s,a,\cdot) - T(s,a,\cdot)||_1 \leq \epsilon$ *and* $|\tilde{R}(s,a) - R(s,a)| \leq \epsilon$, *whenever* $n(s,a) \geq b(\epsilon)$.

Proof: Using the reward and transition confidence intervals, we require that $b(\epsilon) \geq \max\{\frac{8[\ln(2^{|S|}-2)-\ln(\delta_T)]}{\epsilon^2}, \frac{2\ln(2/\delta_R){R_{\max}}^2}{\epsilon^2}\}$. Although δ_T and δ_R depend on $n(s,a)$, we can choose $b(\epsilon) = O\left(\frac{|S|{R_{\max}}^2}{\epsilon^2}\ln\left(\frac{(|S|)(|A|){R_{\max}}^2}{\epsilon^2\delta}\right)\right)$, and still satisfy the required condition. □

We've mentioned that MBIE assumes "optimism in the face of uncertainty", meaning that the expected return of acting in the agent's model is at least as large as the expected return of acting in the underlying environment.

Lemma 7 *Suppose that all confidence intervals computed by MBIE are consistent. Then, for any state* s *and action* a, *the condition* $\tilde{Q}(s,a) \geq Q^*(s,a)$ *is satisfied during any iteration of MBIE.*

Proof: At each step of the learning problem, MBIE solves the MDP $\tilde{M}$. We prove the claim by induction on the number of steps of value iteration. For the base case, assume that the Q values are initialized to $v_{\max} \geq V^*(s)$, for all s. Now, for the induction, suppose that the claim holds for the current value function $\tilde{Q}(s,a)$.

MBIE computes two confidence intervals. $CI(R)$ is an interval of real numbers of the form $(\hat{R}(s,a) - \epsilon^R_{n(s,a)}, \hat{R}(s,a) + \epsilon^R_{n(s,a)})$. $CI(T)$ is the set of probability distributions $T'(s,a,\cdot)$ of the form $||\hat{T}(s,a,\cdot) - T'(s,a,\cdot)||_1 \leq \epsilon^T_{n(s,a)}$. By assumption, we have that $R(s,a) \in CI(R)$ and $T(s,a,\cdot) \in CI(T)$.

The term $\tilde{Q}(s',a')$ on the right-hand side of Equation 6 is the result of the previous iteration and is used to compute the new Q-value $\tilde{Q}(s,a)$ on the left-hand side of the equation. By our confidence-interval assumption, we have $\tilde{R}(s,a) \geq R(s,a)$ and

$$\begin{aligned}
&\max_{\tilde{T}(s,a,\cdot)\in CI(T)} \gamma\sum_{s'}\tilde{T}(s,a,s')\max_{a'}\tilde{Q}(s',a') \\
&\geq \gamma\sum_{s'}T(s,a,s')\max_{a'}\tilde{Q}(s',a') \\
&\geq \gamma\sum_{s'}T(s,a,s')\max_{a'}Q^*(s',a').
\end{aligned}$$

The first step follows from the assumption that $T(s,a,\cdot) \in CI(T)$ and the second from the inductive hypothesis. □

7. Sample Complexity of MBIE

We can now prove that MBIE is PAC in the sample-complexity framework. This section parallels the proof that Rmax has low sample complexity (Kakade, 2003). The main difference in our analysis is that we worked within the discounted reward framework and allowed for MBIE's model to be updated on every step.

At the beginning of a run, every state-action (s,a) pair is said to be *unknown*. At any step of the algorithm, the set of *known* state-action pairs K is defined to be those (s,a) experienced at least m times (Kearns & Singh, 2002). For large m, any $(s,a) \in K$ will be accurately modeled. The concept of a known state is

not used by MBIE—just its analysis. In contrast, E^3 and Rmax explicitly keep track of known states.

An overview of the sample-complexity analysis is as follows. At each timestep, MBIE follows the optimal policy of its model $\tilde{M}$. Lemma 8 shows that the value of MBIE's policy in its model is very close to its true value as long as the probability of reaching an unknown state-action pair is low. By Lemma 7, the estimated value of its policy is at least as large, with high probability, as the true optimal value function. Thus, MBIE chooses its actions based on a policy that is either nearly optimal or one with a high probability of encountering an unknown (s, a). However, the number of times a given (s, a) can be experienced before it becomes known is shown to be no more than polynomial in the relevant quantities. Therefore, the agent will act nearly optimally on all but a bounded number of timesteps—it has polynomial sample complexity.

Lemma 8 (Generalized Induced Inequality) *Let M be an MDP, K a set of state-action pairs, M' an MDP equal to M on K (identical transition and reward functions), π a policy, and T some positive integer. Let A_M be the event that a state-action pair not in K is encountered in a trial generated by starting from state s_1 and following π for T steps in M. Then,*

$$V_M^\pi(s_1, T) \geq V_{M'}^\pi(s_1, T) - v_{\max} \Pr(A_M).$$

Proof: For some fixed *partial path* $p_t = s_1, a_1, r_1 \ldots, s_t, a_t, r_t$, let $P_{t,M}(p_t)$ be the probability p_t resulted from execution of policy π in M starting from state s_1. Let K_t be the set of all paths p_t such that every state-action pair (s_i, a_i) with $1 \leq i \leq t$ appearing in p_t is "known" (in K). Let $r_M(t)$ be the reward received by the agent at time t, and $r_M(p_t, t)$ the reward at time t given that p_t was the partial path generated. Now, we have the following:

$$\begin{aligned}
&E[r_{M'}(t)] - E[r_M(t)] \\
&= \sum_{p_t \in K_t} (P_{t,M'}(p_t) r_{M'}(p_t, t) - P_{t,M}(p_t) r_M(p_t, t)) \\
&\quad + \sum_{p_t \notin K_t} (P_{t,M'}(p_t) r_{M'}(p_t, t) - P_{t,M}(p_t) r_M(p_t, t)) \\
&= \sum_{p_t \notin K_t} (P_{t,M'}(p_t) r_{M'}(p_t, t) - P_{t,M}(p_t) r_M(p_t, t)) \\
&\leq \sum_{p_t \notin K_t} P_{t,M'}(p_t) r_{M'}(p_t, t) = R_{\max} \Pr(A_M).
\end{aligned}$$

The first step in the above derivation involved separating the possible paths in which the agent encounters an unknown state-action from those in which only known state-action pairs are reached. We can then eliminate the first term, because M and M' behave identically on known state-action pairs. The result then follows from that fact that $V_{M'}^\pi(s_1, T) - V_M^\pi(s_1, T) = \sum_{t=0}^{T-1} \gamma^t \left(E[r_{M'}(t)] - E[r_M(t)]\right)$. □

The following proposition states that MBIE is PAC in the sample-complexity framework.

Proposition 3 *Let M be an MDP, $\mathcal{A}_t$ be MBIE's policy at time t, and s_t be the state at time t. With probability at least $1 - \delta$, $V_M^{\mathcal{A}_t}(s_t) \geq V_M^*(s_t) - \epsilon$ is true for all but $O\left(\frac{|S|^2|A|{R_{\max}}^5 \ln^3 \frac{|S|\cdot|A|\cdot R_{\max}}{(1-\gamma)\epsilon\delta}}{(1-\gamma)^6\epsilon^3}\right)$ timesteps t.*

Proof: We assume that all confidence intervals computed by MBIE are consistent, an assumption that holds with probability at least $1 - \delta/2$. We also require $\delta \leq \epsilon / v_{\max}$ (δ can always be polynomially reduced to this value if necessary). At time t, let K be the set of known state-action pairs, specifically, those tried at least m times by the agent. Recall that the agent $\mathcal{A}_t$ chooses its next action by following an optimal policy $\tilde{\pi}$ of MBIE's internal model $\tilde{M}$ at time t. Let M' be the MDP that is equal to M on K and equal to $\tilde{M}$ on $S \times A - K$. We now choose $m = O\left(\frac{|S|{R_{\max}}^4}{(1-\gamma)^4\epsilon_s^2} \ln\left(\frac{(|S|)(|A|){R_{\max}}^4}{(1-\gamma)^4\epsilon_s^2\delta}\right)\right)$, using Lemma 5 and Lemma 6, to ensure that $|V_{M'}^{\tilde{\pi}}(s) - V_{\tilde{M}}^{\tilde{\pi}}(s)| \leq \epsilon_s$ for all s. Using Lemma 3, let $T = O\left(\frac{1}{1-\gamma} \ln \frac{R_{\max}}{\epsilon_d(1-\gamma)}\right)$ be large enough so that for all policies π, $|V_{M'}^\pi(s, T) - V_{M'}^\pi(s)| \leq \epsilon_d$. Let A_M be the event that $\tilde{\pi}$ "escapes" from K in T steps. By Lemma 8, we have that for all states s:

$$V_M^{\mathcal{A}_t}(s, T) \geq V_{M'}^{\mathcal{A}_t}(s, T) - v_{\max} \Pr(A_M).$$

We now consider two mutually exclusive cases. First, suppose that $\Pr(A_M) \geq \frac{\epsilon_1}{v_{\max}}$, meaning that an agent following $\mathcal{A}_t$ will encounter an unknown (s, a) in T steps with probability at least $\frac{\epsilon_1}{v_{\max}}$. Using the Hoeffding bound, after $O(\frac{m|S||A|Tv_{\max}}{\epsilon_1} \ln \frac{1}{\delta_Q})$ timesteps t where $\Pr(A_M) \geq \frac{\epsilon_1}{v_{\max}}$ is satisfied, all (s, a) will become known, with probability at least $1 - \delta_Q$. Now, suppose that $\Pr(A_M) < \frac{\epsilon_1}{v_{\max}}$. Note that $|V_{M'}^{\mathcal{A}_T}(s, T) - V_{M'}^*(s, T)| \leq \epsilon_s + v_{\max} \Pr(A_M) + (\delta/2) v_{\max}$, which can be seen by considering executing policy $\mathcal{A}_T$ in the MDP M' for T steps from state s. As long as the agent encounters only known state-action pairs and it maintains correct confidence intervals, it will, by Lemma 5, achieve ϵ_s-optimal average behavior (recall that MBIE always acts according to some stationary policy $\tilde{\pi}$). Otherwise, the probability that it either encounters an unknown state-action pair or computes an incorrect confidence interval is bounded by $\Pr(A_M) + (\delta/2)$,

yielding

$$\begin{aligned} V_M^{\mathcal{A}_t}(s,T) &\geq V_{M'}^*(s,T) - \epsilon_s - \epsilon_1 - (\delta/2)v_{\max} \\ &\geq V_{M'}^*(s) - \epsilon_s - \epsilon_d - \epsilon_1 - \epsilon/2 \\ &\geq V_{\tilde{M}}^{\tilde{\pi}}(s) - 2\epsilon_s - \epsilon_d - \epsilon_1 - \epsilon/2 \\ &\geq V_M^*(s) - 2\epsilon_s - \epsilon_d - \epsilon_1 - \epsilon/2. \end{aligned}$$

The last step made use of Lemma 7. Thus, if $\delta_Q = \delta/2$ and $\epsilon_d = \epsilon_s = \epsilon_1 = \epsilon/8$, then MBIE's policy is ϵ-optimal with probability at least $1-\delta$ for all but $O(\frac{m|S||A|Tv_{\max}}{\epsilon} \ln \frac{1}{\delta})$ many timesteps. □

The bounds $O\left(\frac{|S|^2|A|{R_{\max}}^5 \ln^3 \frac{|S|\cdot|A|\cdot R_{\max}}{(1-\gamma)\epsilon\delta}}{(1-\gamma)^6\epsilon^3}\right)$ in Proposition 3 are comparable to those achieved by Kearns and Singh (2002) and Kakade (2003) for the algorithms E^3 and Rmax, respectively, especially when modified to account for differences in basic assumptions.

8. Conclusion

Reinforcement-learning algorithms that fully exploit limited and costly real-world experience to maximize their performance are crucial to the future of the field. MBIE takes a step in this direction and, based on recent experimental studies, appears very promising. In comparison to known PAC algorithms, MBIE more smoothly integrates exploration and exploitation.

We've shown that MBIE's worst case PAC bounds are on par with those of E^3 and Rmax. In doing so, we surveyed the progression of PAC concepts from the reset assumption to settings that are increasingly "online" in that are based on state trajectories encountered during learning. We've also discovered that algorithms that are PAC in the sample-complexity setting are also PAC in the average-loss setting.

Our ongoing work attempts to scale MBIE to more realistic domains such as MDPs with continuous or factored state spaces. We are also working on an analysis that will demonstrate that MBIE has a provable advantage over existing PAC learning algorithms in certain classes of MDPs.

Acknowledgments

Thanks to the National Science Foundation (IIS-0325281) and to DARPA IPTO for some early support. We also thank colleagues from the Rutgers RL^3 lab as well as Sanjoy Dasgupta, John Langford, Shie Mannor, David McAllester, Rob Schapire, Rich Sutton, Sergio Verdu, Tsachy Weissman, and Martin Zinkovich for suggestions.

References

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

Fiechter, C.-N. (1997). Expected mistake bound model for on-line reinforcement learning. *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 116–124).

Fong, P. W. L. (1995). A quantitative study of hypothesis selection. *Proceedings of the Twelfth International Conference on Machine Learning (ICML-95)* (pp. 226–234).

Kaelbling, L. P. (1993). *Learning in embedded systems*. Cambridge, MA: The MIT Press.

Kakade, S. M. (2003). *On the sample complexity of reinforcement learning*. Doctoral dissertation, Gatsby Computational Neuroscience Unit, University College London.

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

Puterman, M. L. (1994). *Markov decision processes—discrete stochastic dynamic programming*. New York, NY: John Wiley & Sons, Inc.

Strehl, A. L., & Littman, M. L. (2004). An empirical evaluation of interval estimation for Markov decision processes. *The 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI-2004)* (pp. 128–135).

Strehl, A. L., & Littman, M. L. (2005). A theoretical analysis of model-based interval estimation: Proofs. Forthcoming tech report, Rutgers University.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. The MIT Press.

Voltaire (1759). *Candide*.

Weissman, T., Ordentlich, E., Seroussi, G., Verdu, S., & Weinberger, M. J. (2003). *Inequalities for the L1 deviation of the empirical distribution* (Technical Report HPL-2003-97R1). Hewlett-Packard Labs.

Wiering, M., & Schmidhuber, J. (1998). Efficient model-based exploration. *Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior (SAB'98)* (pp. 223–228).