Action Elimination and Stopping Conditions for Reinforcement Learning

Eyal Even-Dar EVEND@CS.TAU.AC.IL
School of Computer Science, Tel-Aviv University, 69978, Israel

Shie Mannor SHIE@MIT.EDU
Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139

Yishay Mansour MANSOUR@CS.TAU.AC.IL
School of Computer Science, Tel-Aviv University, 69978, Israel

Abstract

We consider incorporating action elimination procedures in reinforcement learning algorithms. We suggest a framework that is based on learning an upper and a lower estimates of the value function or the Q-function and eliminating actions that are not optimal. We provide a model-based and a model-free variants of the elimination method. We further derive stopping conditions that guarantee that the learned policy is approximately optimal with high probability. Simulations demonstrate a considerable speedup and added robustness.

1. Introduction

Reinforcement Learning (RL) has emerged in the recent decade as unified discipline for adaptive control of dynamic environments (e.g., Barto & Sutton, 1998). A common problem with many RL algorithms is a slow convergence rate, even for relatively small problems. For example, consider the popular Q-learning algorithm (Watkins, 1989) which is essentially an asynchronous stochastic approximation algorithm (Bertsekas & Tsitsiklis, 1996). Generic convergence rate bounds for stochastic approximation (e.g., Borkar & Meyn, 2000) or specific rates for Q-learning (see Kearns & Singh, 1998; Even-Dar & Mansour, 2001) are somewhat disappointing.

The problem of finding optimal policies in Markov Decision Processes (MDPs) was the subject of intensive research since the 1950's. When the model is known, and learning is not required there are several standard methods for calculating the optimal policy - Linear Programming, Value Iteration, Policy Iteration etc., see Puterman (1994) for a review. Starting from MacQueen (1966) several algorithms that eliminate actions were proposed. When the MDP model is known Action Elimination (AE) serves two purposes: reduce the size of the action sets to be searched at every iteration; identify optimal policies when there is a unique optimal policy. (In Value Iteration this is the only way to reach optimal policy rather than ϵ-optimal policy.) AE procedures became standard practice in solving large practical MDPs and are considered state-of-the-art. (See Puterman (1994) for more details.) In this paper we consider the learning aspect of AE when the model is *not* known.

In many applications the computational power is available but sampling of the environment is expensive. By eliminating sub-optimal actions early in the learning process, the total amount of sampling is reduced, leading to spending less time on estimating the parameters of sub-optimal actions. The main motivation for applying AE in RL is reducing the amount of samples needed from the environment. In addition to that, AE in RL enjoys the same advantages as in MDPs - convergence rate speedup and possibility to find an optimal policy (rather than ϵ-optimal).

We suggest a framework for AE in RL. The underlying idea is to maintain upper and lower estimates of the value (or Q) function. When the expected upper estimate of the return of a certain action falls below the expected lower estimate of another action, the obviously inferior action is eliminated. We suggest both,

Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

a model-based and a Q-learning style AE algorithms. The upper and lower bounds are based on a large deviations inequality, so that when an action is eliminated, it is eliminated with high probability.

Stopping times that are based on generic convergence rate bounds (as in Even-Dar & Mansour, 2001) are overly conservative. We suggest a stopping time based on the difference between the upper and lower bounds of the value (or Q) function. We show that if the difference is small, then the greedy policy with respect to the lower estimate is almost optimal.

2. The Model

We define a Markov Decision process (MDP) as follows

Definition 2.1 *A Markov Decision process (MDP) M is a 4-tuple (S, A, P, R), where S is a set of the states, A is a set of actions, $P^a_{i,j}$ is the transition probability from state i to state j when performing action $a \in A$ in state i, and $R(s,a)$ is the reward received when performing action a in state s.*

A strategy for an MDP assigns, at each time t, for each state s a probability for performing action $a \in A$, given a history $F_{t-1} = \{s_1, a_1, r_1, ..., s_{t-1}, a_{t-1}, r_{t-1}\}$ which includes the states, actions and rewards observed until time $t-1$. While executing a strategy π we perform at time t action a_t in state s_t and observe a reward r_t (distributed according to $R(s,a)$), and the next state s_{t+1} distributed according to $P^{a_t}_{s_t,s_{t+1}}$. We combine the sequence of rewards into a single value called the *return*. Our goal is to maximize the return. In this work we focus on the *discounted return*, which has a parameter $\gamma \in (0,1)$, and the discounted return of policy π is $V^\pi = \sum_{t=0}^{\infty} \gamma^t r_t$, where r_t is the reward observed at time t. We also consider the *finite horizon* return, $V^\pi = \sum_{t=0}^{H} r_t$ for a given horizon H.

We assume that $R(s,a)$ is non-negative and bounded by R_{max}, i.e., for every $s, a: \quad 0 \le R(s,a) \le R_{max}$ and for simplicity we assume that $R(s,a)$ is deterministic and note that all the results apply for stochastic rewards as well (under minor changes in the proof). This implies that the discounted return is bounded by $V_{max} = \frac{R_{max}}{1-\gamma}$; for the finite horizon the return is bounded by HR_{max}. We define a value function for each state s, under policy π, as $V^\pi(s) = \mathbf{E}^\pi[\sum_{i=0}^{\infty} r_i \gamma^i]$, where the expectation is over a run of policy π starting at state s, and a state-action value function $Q^\pi(s,a) = R(s,a) + \gamma \sum_{s'} P^a_{s,s'} V^\pi(s')$. Similarly, we define the value functions for the finite horizon model.

Let π^* be an optimal policy which maximizes the return from any start state. This implies that for any policy π and any state s we have $V^{\pi^*}(s) \ge V^\pi(s)$, and $\pi^*(s) = \text{argmax}_a(R(s,a) + \gamma(\sum_{s'} P^a_{s,s'} V^{\pi^*}(s'))$. We use V^* and Q^* for V^{π^*} and Q^{π^*}, respectively. We say that a policy π is ϵ-optimal if $\|V^* - V^\pi\|_\infty \le \epsilon$. We also define the policy $Greedy(Q)$ as the policy that prescribes in each state the action that maximizes the Q-function in the state, i.e., $\pi(s) = \text{argmax}_a\, Q(s,a)$.

For a given trajectory let: $T^{s,a}$ be the set of times in which we perform action a in state s and $T^{s,a,s'}$ be a subset of $T^{s,a}$ in which we reached state s'. Also, $\#(s,a,t)$ is the number of times action a is performed in state s up to time t, i.e., $|T^{s,a} \cap \{1,2,3,\ldots,t\}|$. Next we define the empirical model. Given that $|T^{s,a}| > 0$ we define the next state distribution as $\hat{P}^a_{s,s'} = \frac{|T^{s,a,s'}|}{|T^{s,a}|}$ and since the reward is deterministic we have that $\hat{R}(s,a) = R(s,a)$. If $|T^{s,a}| = 0$ the empirical model and the reward can be chosen arbitrarily. We define the expectation of the empirical model as $\hat{\mathbf{E}}_{s,s',a}[V(s')] = \sum_{s' \in S} \hat{P}^a_{s,s'} V(s')$. To simplify the notations we omit s, a in the notations $\hat{\mathbf{E}}_{s'}$ whenever evident.

We often use large deviation bounds in this paper. Since we assume boundedness we can rely on Hoeffding's inequality (We note that the boundedness assumption is not essential and can be relaxed.)

Lemma 2.1 *(Hoeffding, 1963) Let X be a set, D be a probability distribution on X, and $f_1, ..., f_m$ be real-valued functions defined on X with $f_i : X \to [a_i, b_i]$ for $i = 1, ..., m$, where a_i and b_i are real numbers satisfying $a_i < b_i$. Let $x_1, \ldots, x_m$ be IID samples from D. Then we have the following inequality*

$$\mathbf{P}\left[\left|\frac{1}{m}\sum_{i=1}^{m} f_i(x_i) - \left(\frac{1}{m}\sum_{i=1}^{m}\int_{a_i}^{b_i} f_i(x)D(x)\right)\right| \ge \epsilon\right] \le 2e^{-2\epsilon^2 m^2 \frac{1}{\sum_{i=1}^{m}(b_i-a_i)^2}}.$$

3. Model-Based Learning

In this section we focus on model-based learning. In the model-based methods, we first learn the model, i.e., estimate the immediate reward and the next state distribution. Then by either value iteration or policy iteration on the learned (empirical) model, we find the exact optimal policy for the empirical model. If enough exploration is done, this policy is an almost optimal policy for the real model. We note that there is an inherent difference between the finite horizon and the infinite discounted return. Technically, the finite

horizon return is simpler than the discounted return, as one can apply the large deviation bounds directly. We provide model-based algorithms for both cases.

3.1. Finite Horizon

Let us first recall the classical Value Iteration equations for finite horizon:

$$\begin{aligned} V^H(s) &= \max_a\{R(s,a)+\mathbf{E}_{s'}[V^{H-1}(s')]\}, \quad H>0 \\ V^0(s) &= \max_a R(s,a), \end{aligned}$$

where $V^H(s)$ is the optimal value function for horizon H. Given the empirical model by time t we define the upper estimate $\overline{V}_\delta$, which will be shown to satisfy for every horizon k and every state s, $\overline{V}^k_\delta(s) \geq V^k(s)$ with high probability. For horizon H we define:

$$\begin{aligned} \overline{V}^H_\delta(s) &= \max_a\Big\{R(s,a)+\hat{\mathbf{E}}_{s'}[\overline{V}^{H-1}_\delta(s')]+ \qquad (1) \\ &\qquad HR_{max}\sqrt{\frac{\ln(\frac{c|S||A|H^2}{\delta})}{|T^{s,a}|}}\Big\}, \quad H>0 \\ \overline{V}^0_\delta(s) &= \max_a R(s,a), \qquad (2) \end{aligned}$$

for some constant $c>2$. Similarly to the upper bound $\overline{V}^H_\delta$, a lower bound may be defined where the plus sign before the last element of Eq. (1) is replaced by a minus sign. We call this estimate the lower estimate $\underline{V}^H_\delta$. The following Lemma proves that $\overline{V}^H_\delta$ ($\underline{V}^H_\delta$) is indeed an upper (lower) estimation for any horizon. (The proof appears in Appendix A.)

Lemma 3.1 *Every state s and for every finite horizon k, we have that $\overline{V}^k_\delta(s) \geq V^k(s) \geq \underline{V}^k_\delta(s)$ with probability at least $1-\delta$.*

Consequently, a natural early stopping condition is to stop sampling when $\|\overline{V}^H - \underline{V}^H\|_\infty < \epsilon$. We do not provide here an algorithm, however a detailed algorithm will be given in the following subsection.

3.2. Discounted Return - Infinite Horizon

In this subsection, we provide an upper estimate of the value function V. The optimal value is the solution of the set of the equations:

$$V^*(s) = \max_a\{R(s,a)+\gamma\mathbf{E}_{s'}[V^*(s')]\}, \quad s\in S.$$

As in Subsection 3.1, we provide an upper value function $\overline{V}_\delta$, which satisfies with high probability $\overline{V}_\delta(s) \geq V^*(s)$. We define $\overline{V}_\delta$ as the solution of the set of equations:

$$\overline{V}_\delta(s) = \max_a\Big\{R(s,a)+\gamma\hat{\mathbf{E}}_{s'}[\overline{V}_\delta(s')]+V_{max}\sqrt{\frac{\ln(\frac{2|S|\,|A|}{\delta})}{|T^{s,a}|)}}\Big\}$$

and $\overline{Q}_\delta$ as:

$$\overline{Q}_\delta(s,a) = R(s,a)+\gamma\hat{\mathbf{E}}_{s'}[\overline{V}_\delta(s')]+V_{max}\sqrt{\frac{\ln(\frac{2|S|\,|A|}{\delta})}{|T^{s,a}|}}.$$

Similarly, we define $\underline{V}_\delta$ and $\underline{Q}_\delta$ as:

$$\underline{V}_\delta(s) = \max_a\Big\{R(s,a)+\gamma\hat{\mathbf{E}}_{s'}\underline{V}_\delta(s')-V_{max}\sqrt{\frac{\ln(\frac{2|S|\,|A|}{\delta})}{|T^{s,a}|)}}\Big\}$$

$$\underline{Q}_\delta(s,a) = R(s,a)+\gamma\hat{\mathbf{E}}_{s'}[\underline{V}_\delta(s')]-V_{max}\sqrt{\frac{\ln(\frac{2|S|\,|A|}{\delta})}{|T^{s,a}|}}.$$

The next lemma shows that with high probability the upper and lower estimations are indeed correct. (The proof is deferred to Appendix B.)

Lemma 3.2 *For every state s and action a with probability at least $1-\delta$ we have that $\overline{Q}_\delta(s,a) \geq Q^*(s,a) \geq \underline{Q}_\delta(s,a)$.*

The AE procedure is demonstrated in the following algorithm, which supplies a stopping condition for sampling the model and eliminates actions when they are clearly sub-optimal.

Input : MDP M, $\epsilon>0$, $\delta>0$
Output: A policy for M
Choose arbitrarily an initial state s_0, let $t=0$, and let $U_0=\{(s,a)|s\in S, a\in A\}$
repeat
  At state s_t perform any action a s.t. $(s_t,a)\in U_t$
  Receive a reward r_t, and a next state s_{t+1}
  Compute, $\overline{Q}_\delta, \underline{Q}_\delta$ from all the samples
  $t=t+1$
  $U_t=\{(s,a)|\overline{Q}_\delta(s,a)\geq \underline{V}_\delta(s)\}$
until $\forall(s,a)\in U \quad |\overline{Q}_\delta(s,a)-\underline{Q}_\delta(s,a)|<\frac{\epsilon(1-\gamma)}{2}$;
return $Greedy(\underline{Q}_\delta)$

Algorithm 1: Model-Based AE Algorithm

A direct corollary from Lemma 3.2, is a stopping time condition to the Model-Based algorithm using the following Corollary.

Corollary 3.3 *(Singh & Yee, 1994) If $\tilde{Q}$ is a function such that $|\tilde{Q}(s,a)-Q^*(s,a)|\leq\epsilon$ for all $s\in S$ and $a\in A$. Then for all s*

$$V^*(s)-V^{\tilde{\pi}}(s)\leq\frac{2\epsilon}{1-\gamma},$$

where $\tilde{\pi}=Greedy(\tilde{Q})$.

Corollary 3.4 *Supposed the Model-Based AE Algorithm terminates. Then the policy, π, it returns is ϵ-optimal with probability at least $1-\delta$.*

Proof: By Lemma 3.2 we know that with probability at least $1-\delta$ for every s and a we have that $\underline{Q}_\delta(s,a) \leq Q^*(s,a) \leq \overline{Q}_\delta(s,a)$. Therefore, with probability of at least $1-\delta$ the optimal action has not been eliminated in any state. Furthermore, any action b in state s that has not been eliminated satisfies $Q^*(s,b) - \underline{Q}_\delta(s,b) \leq \overline{Q}_\delta(s,b) - \underline{Q}_\delta(s,b) \leq \frac{\epsilon(1-\gamma)}{2}$. The result follows by Corollary 3.3. ∎

4. Model-Free Learning

In this section we describe a model-free algorithm. We use two functions $\underline{Q}_t$ and $\overline{Q}_t$, which provide lower and upper estimations on Q^*, respectively. We use these functions to derive an asynchronous algorithm, which eliminates actions and supplies stopping condition. Let us first recall the Q-learning algorithm (Watkins, 1989). This algorithm requires space which is proportional to the space used by Q-learning and converges under the same conditions. The Q-learning algorithm estimates the state-action value function (for discounted return) as follows:

$$\begin{aligned} Q^0(s,a) &= 0, \\ Q^{t+1}(s,a) &= (1-\alpha_t(s,a))Q^t(s,a) + \\ &\quad \alpha_t(s,a)(r_t(s,a) + \gamma V^t(s')), \end{aligned}$$

where s' is the state reached from state s when performing action a at time t, and $V^t(s) = \max_a Q^t(s,a)$. Set $\alpha_t(s,a) = \frac{1}{\#(s,a,t)}$ for $t \in T^{s',a'}$ and 0 otherwise. We define the upper estimation process as:

$$\begin{aligned} \overline{Q}^0_\delta(s,a) &= V_{max} \ln(\frac{c|S||A|}{\delta}), \\ \overline{Q}^{t+1}_\delta(s,a) &= (1-\alpha_t(s,a))\overline{Q}^t_\delta(s,a) + \alpha_t(s,a) \cdot \\ &\quad \Big(R(s,a) + \gamma \overline{V}^t_\delta(s') + \beta(\#(s,a,t))\Big), \end{aligned}$$

where $c > 4$ and s' is the state reached from state s when performing action a at time t, $\overline{V}^t_\delta(s) = \max_a \overline{Q}^t_\delta(s,a)$ and

$$\beta(k) = \sqrt{\frac{\ln(ck^2|S||A|/\delta)}{k}}.$$

Analogously, we define the lower estimate $\underline{Q}_\delta$ as :

$$\begin{aligned} \underline{Q}^0_\delta(s,a) &= -V_{max} \ln(\frac{c|S||A|}{\delta}), \\ \underline{Q}^{t+1}_\delta(s,a) &= (1-\alpha_t(s,a))\underline{Q}^t_\delta(s,a) + \alpha_t(s,a) \\ &\quad \Big(R(s,a) + \gamma \underline{V}^t_\delta(s') - \beta(\#(s,a,t))\Big), \end{aligned}$$

Input : MDP M, $\epsilon > 0$, $\delta > 0$
Output: A policy for M
For every state action (s,a):
$\overline{Q}(s,a) = \quad V_{max} \ln(\frac{c|S||A|}{\delta})$
$\underline{Q}(s,a) = -V_{max} \ln(\frac{c|S||A|}{\delta})$
$\#$(s,a) = 1
Choose an arbitrary initial state s
repeat
  Let $U(s) = \{a | \overline{Q}(s,a) \geq \underline{V}(s)\}$
  choose arbitrarily action $a \in U(s)$, perform it and observe the next state s'
  $\overline{Q}(s,a) := (1 - \frac{1}{\#(s,a)})\overline{Q}(s,a) + \frac{1}{\#(s,a)}\Big(R(s,a) + \gamma\overline{V}(s') + \beta(\#(s,a))\Big)$
  $\underline{Q}(s,a) := (1 - \frac{1}{\#(s,a)})\underline{Q}(s,a) + \frac{1}{\#(s,a)}\Big(R(s,a) + \gamma\underline{V}(s') - \beta(\#(s,a))\Big)$
  $\#(s,a) := \#(s,a) + 1$; $s = s'$
until $\forall s \in S \ \ \forall a \in U(s) \quad |\overline{Q}(s,a) - \underline{Q}(s,a)| < \frac{\epsilon(1-\gamma)}{2}$;
return *Greedy*($\underline{Q}$)

Algorithm 2: Model-Free AE Algorithm

where $\underline{V}_\delta(s) = \max_a \underline{Q}_\delta(s,a)$. We claim that these processes converge almost surely to Q^*. (The proof appears in Appendix C.)

Proposition 4.1 *If every state-action pair is performed infinitely often then the upper (lower) estimation process, $\overline{Q}^t_\delta$ ($\underline{Q}^t_\delta$), converges to Q^* with probability one.*

The following Proposition claims that $\overline{Q}^t_\delta$ upper bounds Q^* and $\underline{Q}^t_\delta$ lower bounds Q^* with high probability. (The proof appears in Appendix D.)

Proposition 4.2 *For every state action pair s,a and time t with probability at least $1-\delta$ we have that $\overline{Q}^t_\delta(s) \geq Q^*(s) \geq \underline{Q}^t_\delta(s)$.*

We combine the upper and lower estimates to an algorithm, which eliminates sub optimal actions whenever possible. Furthermore, the algorithm supplies a stopping condition that assures a near optimal policy. The model free AE algorithm is described in Algorithm 2.

A direct corollary from Proposition 4.2 is a stopping condition to the model free AE algorithm. The following corollary follows from Corollary 3.3 and its proof is similar to the proof of Corollary 3.4.

Corollary 4.3 *Suppose the Model-Free AE Algorithm terminates. Then the policy, π, it returns is ϵ-optimal with probability at least $1-\delta$.*

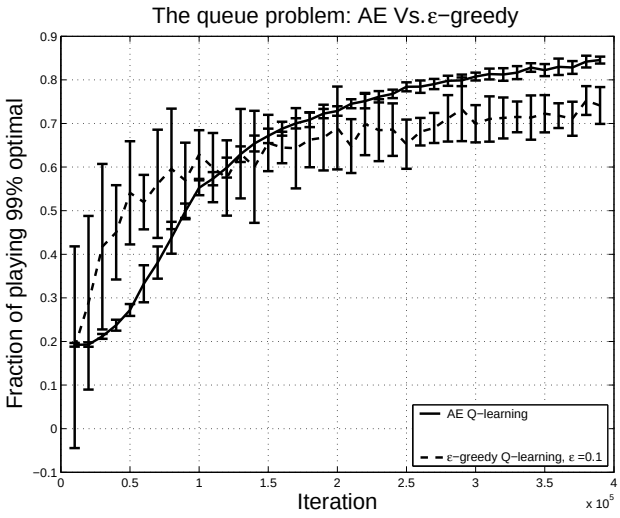

Figure 1. Example of a Queue of size 5 with three types of packets with values $1, 20, 150$. The discount factor is set to 0.99. We disregard the full queue state in which every action is optimal. We repeated each experiment 15 times and the error bars represent 1 standard deviation.

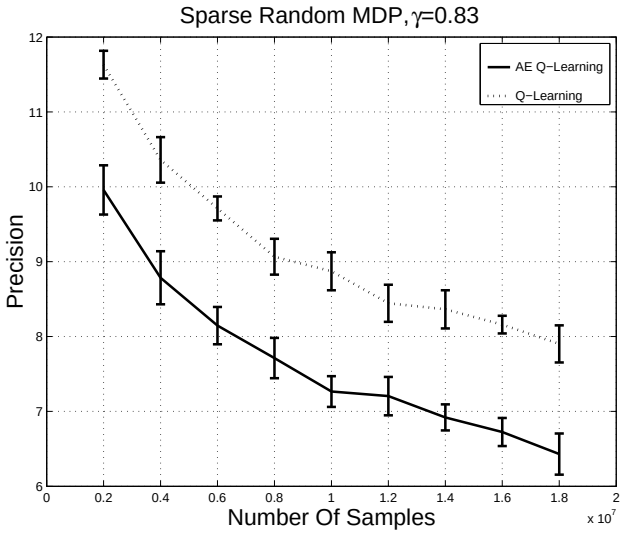

Figure 2. Example of a 20 state sparse randomly generated MDPs with 50 actions in each state, where $\gamma = 0.833$ (as in Puterman, 1994.) The precision is the distance of the Q-function from the optimal Q-function. We repeated each experiment 10 times and the error bars represent 1 standard deviation.

5. Experiments

In this section we show four types of MDPs in which the number of samples used by AE procedures is significantly smaller than the number of samples used by standard Q-learning and ϵ-greedy Q-learning. Both model free AE algorithm and standard Q-learning choose the action in each state uniformly at random. In our experiments we focused on the steady state norm (L_1 weighted by steady state probabilities) rather than the L_∞ norm. We note that we use the steady state rather than the discounted steady state. We run AE Q-learning algorithm from Section 4 with the same input (for actions that were not eliminated) as a standard Q-learning algorithm. The following experiments were conducted:

1. **A queueing system.** The MDP represents a queueing problem that appears in Differentiated Services (Aiello et al., 2000; Kesselman et al., 2001). The basic settings are that the arriving packets have different values and they are buffered in a FIFO queue to be sent. The major constraints are that we reject or accept a packet upon its arrival (no preemption) and that the buffer has limited capacity. We have looked on a queue of size five and three different packets values, $1, 20, 150$. In each time unit we either receive a packet or send a packet according to some distribution. We modelled the queueing problem via a discounted MDP with discount factor $\gamma = 0.99$. The AE model free algorithm was compared with epsilon greedy Q-learning with epsilon varying from 0.05 to 0.2. In Figure 1 we present the results for ϵ which was empirically best, $\epsilon = 0.1$. In this experiment we used a fixed step size. Rather than looking on distance from the optimal value, we focused here on the fraction of times in which optimal actions were performed. The results are demonstrated in Figure 1, in which we see that not only AE has better results but the variance in the results is much smaller. While not shown in Figure 1, the AE was superior in terms of the value function as well. (The output policy of the AE Q-learning algorithm achieved on average 85% of the optimal value function, while the ϵ-greedy output policy achieved about 70% of the optimal value function.)

2. **Random MDPs.** The random MDPs are randomly generated MDPs with 20 states each and 50 actions in each state. The first one is due to Puterman (1994) and is a sparse MDP, such that each action can reach only three states. The second random MDP is a dense MDP, such that the next state distribution is randomly chosen for each state-action pair and might include all states. For both MDPs the immediate reward expectation is randomly chosen in the interval $[0, 10]$. Results of ten runs are presented by Figure 2 for the sparse MDP, in this experiment the model free

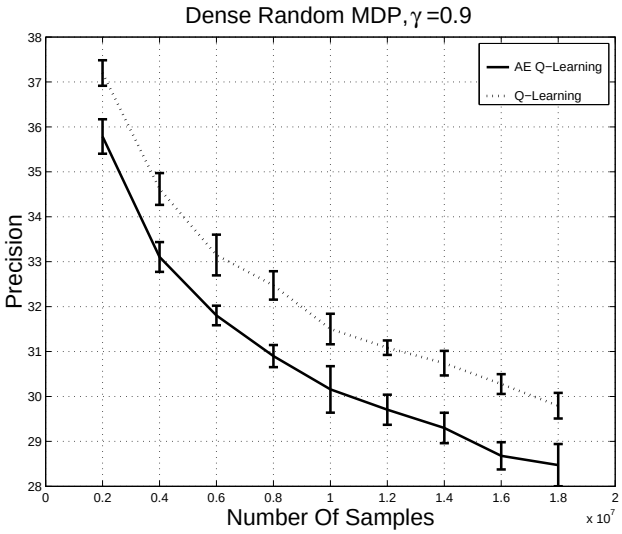

Figure 3. Example of a 20 state dense randomly generated MDPs with 50 actions in each state, $\gamma = 0.9$. The error bars represent 1 standard deviation.

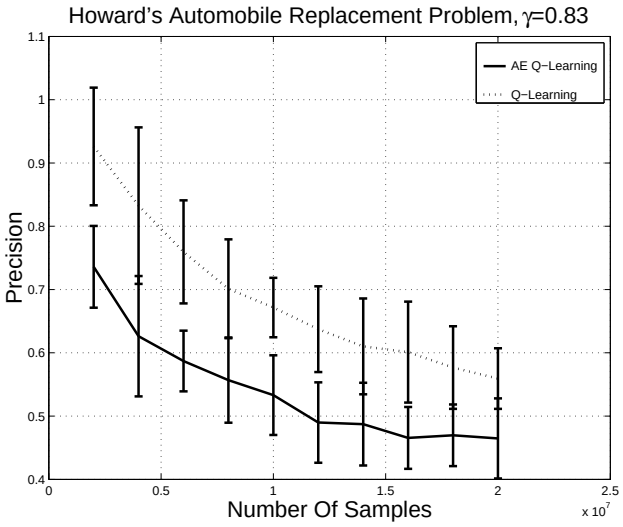

Figure 4. Example of Howard's Automobile Replacement Problem, where the discount factor, γ, is 0.833. The norm is the steady state norm. The error bars represent 1 standard deviation.

AE algorithm needs only about half the samples used by the Q-learning to achieve the same precision. The precision is measured as the distance of the Q-function from the optimal function in steady state norm. In Figure 3 for dense MDP, the results are similar. The AE algorithm required about 40% fewer samples.

3. **Howard's automobile replacement problem.** This MDP represents another realistic problem—Howard's automobile replacement problem (Howard, 1960). This problem contains 40 states, in each state there are 41 actions. See Howard (1960) for a detailed description. This problem was considered as a benchmark by several authors in the optimization community. We run the model free AE algorithm for this problem with discount factor $\gamma = 0.833$ against standard Q-learning and the results appear in Figure 4. A significant improvement is evident.

6. Future Directions

Extending the concept of action elimination to large state spaces is probably the most important direction. The extension to function approximation, which approximate the value function, requires some assumptions on the value (or Q) function approximation architecture. Following Kakade and Langford (2002) we can consider value functions that can be approximated under the infinity norm. For an example of such an algorithm see Ormoneit and Sen (2002). If convergence rate of the function approximation is provided, as in Ormoneit and Sen (2002), then an AE procedure can be derived as before.

Acknowledgements

This research was supported in part by a grant from the Israel Science Foundation. S.M. was partially supported by the MIT-Merrill Lynch Partnership. E.E was partially supported by the Deutsch Institute.

References

Aiello, W. A., Mansour, Y., Rajagopolan, S., & Rosen, A. (2000). Competitive queue policies for diffrentiated services. *In INFOCOM*.

Barto, A., & Sutton, R. (1998). *Reinforcement learning*. MIT Press.

Bertsekas, D. P., & Tsitsiklis, J. N. (1996). *Neuro-dynamic programming*. Belmont, MA: Athena Scientific.

Borkar, V., & Meyn, S. (2000). The O.D.E. method for convergence of stochastic approximation and reinforcement learning. *SIAM J. Control Optim.*, *38*, 447–469.

Even-Dar, E., & Mansour, Y. (2001). Learning rates for Q-learning. *Fourteenth Annual Conference on Computation Learning Theory* (pp. 589–604).

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, *58*, 13–30.

Howard, R. (1960). *Dynamic programming and Markov decision processes*. MIT press.

Kakade, S., & Langford, J. (2002). Approximately optimal approximate reinforcement learning. *Proceedings of the Nineteenth International Conference on Machine Learning* (pp. 267–274). Morgan Kaufmann.

Kearns, M., & Singh, S. P. (1998). Finite-sample convergence rates for Q-learning and indirect algorithms. *Neural Information Processing Systems 10* (pp. 996–1002).

Kesselman, A., Lotker, Z., Mansour, Y., Patt-Shamir, B., Schieber, B., & Sviridenko, M. (2001). Buffer overflow management in qos switches. *"ACM Symposium on Theory of Computing"* (pp. 520–529).

MacQueen, J. (1966). A modified dynamic programming method for Markov decision problems. *J. Math. Anal. Appl.*, *14*, 38–43.

Ormoneit, D., & Sen, S. (2002). Kernel-based reinforcement learning. *Machine Learning*, *49*, 161–178.

Puterman, M. (1994). *Markov decision processes*. Wiley-Interscience.

Singh, S. P., & Yee, R. C. (1994). An upper bound on the loss from approximate optimal-value functions. *Machine Learning*, *16*, 227–233.

Watkins, C. (1989). *Learning from delayed rewards*. Doctoral dissertation, Cambridge University.

Appendix

A. Proof of Lemma 3.1

We prove the claim by induction. For the base of the induction we have that for every state s $\overline{V}^0_\delta(s) = V^0(s) = R(s,a)$. Next we assume that the claim holds for $i \leq k$ and prove for $k+1$ and for every action a. By definition $\overline{V}^{k+1}_\delta(s)$ satisfies for every a that

$$\begin{aligned}\overline{V}^{k+1}_\delta(s) &\geq R(s,a) + \hat{\mathbf{E}}_{s'}[\overline{V}^k_\delta(s')] + kR_{max}\sqrt{\frac{\ln(\frac{c|S||A|k^2}{\delta})}{|T^{s,a}|}} \\ &\geq R(s,a) + \hat{\mathbf{E}}_{s'}[V^k(s')] + kR_{max}\sqrt{\frac{\ln(\frac{c|S||A|k^2}{\delta})}{|T^{s,a}|}},\end{aligned}$$

where the second inequality follows from the inductive assumption. Note that V^k is not a random variable, so we can bound the last expression using Hoeffding's inequality. We arrive at:

$$\mathbf{P}\left\{\hat{\mathbf{E}}_{s'}[V^k(s')] + kR_{max}\sqrt{\frac{\ln(\frac{c|S||A|k^2}{\delta})}{|T^{s,a}|}} < \mathbf{E}_{s'}[V^k(s')]\right\} \leq e^{\frac{-\ln(\frac{c|S||A|k^2}{\delta})|T^{s,a}|\left(\frac{kR_{max}}{\sqrt{|T^{s,a}|}}\right)^2}{(kR_{max})^2}} = \frac{\delta}{c|S||A|k^2}.$$

Therefore, we have that with high probability the following holds

$$\begin{aligned}\overline{V}^{k+1}_\delta(s) &\geq R(s,a) + \hat{\mathbf{E}}_{s'}[V^k(s')] + kR_{max}\sqrt{\frac{\ln(\frac{c|S||A|k^2}{\delta})}{|T^{s,a}|}} \\ &\geq R(s,a) + \mathbf{E}_{s'}[V^k(s')] \\ &= V^{k+1}(s').\end{aligned}$$

Using the union bound over all state-action pairs and all finite horizons k, we obtain that the failure probability is bounded by $\delta/2$ for large enough c. Repeating the same argument for the lower estimate and applying the union bound completes the proof. ∎

B. Proof of Lemma 3.2

Suppose we run a value iteration algorithm on the empirical model. Let $\overline{V}^k_\delta$ be the kth iteration of the value function algorithm, and let $\overline{Q}^k_\delta$ be the associated Q-function, that is $\overline{Q}^k_\delta(s,a) = R(s,a) + \gamma\hat{\mathbf{E}}_{s'}[\overline{V}^k_\delta(s')] + V_{max}\sqrt{\frac{\ln(\frac{2|S||A|}{\delta})}{T^{s,a}}}$. Assume that we start with $\overline{V}^0_\delta = V^*$. (The use of V^* is restricted to the proof and not used in the algorithm.) We need to prove that $\overline{Q}_\delta(s,a) \geq Q^*(s,a)$ for every s and a. Note that since the value iteration converges, $\overline{Q}^k_\delta$ converges to $\overline{Q}_\delta$. We prove by induction on the number of the iterations that if we take $\overline{V}^0_\delta = V^*$ then with high probability for every k we have that $\overline{Q}^k_\delta \geq \overline{Q}^{k-1}_\delta$, i.e. $\mathbf{P}[\forall k\ \overline{Q}^k_\delta \geq \overline{Q}^{k-1}_\delta] \geq 1 - \delta/2$. For the basis, since V^* is not a random variable we can apply Hoeffding's inequality and obtain that for every state action pair s, a

$$\mathbf{P}\Big\{\hat{\mathbf{E}}_{s'}[V^*(s')] + V_{max}\sqrt{\frac{\ln(\frac{2|S||A|}{\delta})}{|T^{s,a}|}} < \mathbf{E}_{s'}[V^*(s')]\Big\} \leq e^{-\ln(\frac{2|S||A|}{\delta})} = \frac{\delta}{2|S||A|},$$

Since $\overline{V}_\delta^0(s) = V^*$ we have that $\overline{Q}_\delta^1(s,a) = R(s,a) + \gamma\hat{\mathbf{E}}_{s'}[\overline{V}_\delta^0(s')] + V_{max}\sqrt{\frac{\ln(\frac{2|S||A|}{\delta})}{|T^{s,a}|}}$. Therefore, $\overline{Q}_\delta^1 \geq \overline{Q}_\delta^0$ with probability $1-\delta/2$. For the induction step, we assume that the claim holds for $i < k$ and prove for k.

$$\overline{Q}_\delta^k(s,a) - \overline{Q}_\delta^{k-1}(s,a) \quad = \quad \gamma\hat{\mathbf{E}}_{s'}[\overline{V}_\delta^{k-1}(s') - \overline{V}_\delta^{k-2}(s')].$$

Since $\overline{V}_\delta^{k-1}(s') = \max_a \overline{Q}_\delta^{k-1}(s',a)$ we have by the induction that for every s,

$$V_\delta^{k-1}(s) = \max_a \overline{Q}_\delta^{k-1}(s,a) \geq \max_a \overline{Q}_\delta^{k-2}(s,a) = V_\delta^{k-2}(s).$$

So that $\overline{Q}_\delta^k - \overline{Q}_\delta^{k-1} \geq 0$. We conclude that $\mathbf{P}[\overline{Q}_\delta \geq Q^*] \geq 1-\delta/2$. Repeating the same argument for the lower estimate, $\underline{Q}_\delta$, and applying the union bound completes the proof. ∎

C. Proof of Proposition 4.1

In order to show the almost sure convergence of the upper and lower estimations, we follow the proof of Bertsekas and Tsitsiklis (1996). We consider a general type of *iterative stochastic algorithms*, which is performed as follows:

$$X_{t+1}(i) = (1-\alpha_t(i))X_t(i) + \alpha_t(i)((HX_t)(i) + w_t(i) + u_t(i)),$$

where w_t is a bounded random variable with zero expectation and each H is a pseudo contraction mapping (See Bertsekas and Tsitsiklis (1996) for details).

Definition C.1 *An iterative stochastic algorithm is well behaved if:*

1. *The step size $\alpha_t(i)$ satisfies (1) $\sum_{t=0}^\infty \alpha_t(i) = \infty$, (2) $\sum_{t=0}^\infty \alpha_t^2(i) < \infty$ and (3) $\alpha_t(i) \in (0,1)$.*
2. *There exists a constant A that bounds $w_t(i)$ for any history F_t, i.e., $\forall t, i: \ |w_t(i)| \leq A$.*
3. *There exists $\gamma \in [0,1)$ and a vector X^* such that for any X we have $||HX - X^*|| \leq \gamma||X - X^*||$, where $||\cdot||$ is any norm.*
4. *There exists a nonnegative random sequence θ_t, that converges to zero with probability 1, and is such that*

$$\forall i,t \quad |u_t(i)| \leq \theta_t(||X_t|| + 1)$$

We first note that the Q-learning algorithm satisfies the first three criteria and the fourth criteria holds trivially since $u_t = 0$, thus its convergence follows (see Proposition 5.6 in Bertsekas & Tsitsiklis, 1996). The upper estimate has an additional noise term, u_t. If we show that it satisfies the fourth requirement, then the convergence will follow.

Lemma C.1 *The upper estimation algorithm is well behaved.*

Proof: In the convergence proof of Q-learning, it was shown that requirements 1–3 are satisfied, this implies that the upper estimates satisfies them as well. Now we let $u_t = \theta_t = c\sqrt{\frac{\ln(\#(s,a,t))}{\#(s,a,t)}}V_{max}$. θ_t clearly converges to zero, thus

$$|u_t(i)| = \theta_t \leq \theta_t(||X_t|| + 1).$$

∎

Similar result holds for the lower estimate as well.

D. Proof of Proposition 4.2

We use induction on the time t to show that for every state-action pair the following holds

$$\mathbf{P}\left\{\forall t' \leq t \ \ \overline{Q}_\delta^{t'}(s,a) < Q^*(s,a)\right\} \leq \sum_{i=1}^{\#(s,a,t)} \frac{\delta}{c|S||A|i^2}.$$

For the induction basis we have that for every state-action pair :$\overline{Q}_\delta^1(s,a) \geq Q^*(s,a)$. We assume that the claim holds for $k < t$ and prove for t. Suppose that action a is performed at state s at time t, thus for every other state-action pair the claim holds. Let t_i be the time were action a was performed at state s for the ith time. Let s_i be the next state at time $t_i + 1$. For s, a we have that,

$$\begin{aligned}
\overline{Q}_\delta^t(s,a) &= \frac{1}{\#(s,a,t)} \sum_{i=1}^{\#(s,a,t)} (R(s,a) + \gamma\overline{V}_\delta^{t_i}(s_i) + \beta(i)) \\
&\geq \frac{1}{\#(s,a,t)} \sum_{i=1}^{\#(s,a,t)} (R(s,a) + \gamma V^*(s_i)) \\
&\quad + \sqrt{\frac{\ln(c\#(s,a,t)^2|S||A|/\delta)V_{max}}{\#(s,a,t)}},
\end{aligned}$$

where the last expression can be bounded with high probability using Hoeffding's inequality

$$\begin{aligned}
&\mathbf{P}\Big\{\frac{1}{\#(s,a,t)} \sum_{i=1}^{\#(s,a,t)} (R(s,a) + \gamma V^*(s_i)) + \\
&\qquad \sqrt{\frac{\ln(c\#(s,a,t)^2|S||A|/\delta)V_{max}}{\#(s,a,t)}} < Q^*(s,a)\Big\} \\
&\leq \frac{\delta}{c|S||A|\#(s,a,t)^2},
\end{aligned}$$

which completes the induction by using the union bound. Repeating the same argument for the lower estimate completes the proof. ∎