Boundedness of Iterates in Q-Learning

Abhijit Gosavi

Department of Industrial Engineering

The State University of New York at Buffalo

317 Bell Hall Box 602050, Buffalo, NY 14260-2050

Telephone: (716) 645-2357, Email:agosavi@buffalo.edu

Abstract

Reinforcement Learning (RL) is a simulation-based counterpart of stochastic dynamic programming. In recent years, it has been used in solving complex Markov decision problems (MDPs). Watkins' Q-Learning is by far the most popular RL algorithm used for solving discounted-reward MDPs. The boundedness of the iterates in Q-Learning plays a critical role in its convergence analysis and in making the algorithm stable, which makes it extremely attractive in numerical solutions. Previous results show boundedness asymptotically in an almost sure sense. We present a new result that shows boundedness in an *absolute sense* under some weaker conditions for the step size. Also, our proof is based on some simple induction arguments.

Keywords: Q-Learning, boundedness, stochastic control

1 Introduction

The Markov decision problem (MDP) [3] is a well-known problem in stochastic optimization for which Dynamic Programming (DP) can be used for solution purposes. In complex systems with several governing random variables, oftentimes it is difficult to obtain the values of the so-called transition probabilities required by DP. Reinforcement Learning (RL) [5, 1] is a simulation-based version of DP that can avoid the computation of the transition probabilities. A great deal of interest in recent times in RL has stemmed from its ability to generate near-optimal solutions. Empirical work has now been backed by rigorous theoretical analysis of its convergence (see Borkar and Meyn [2] and Tsitsiklis [6]) placing the method on solid ground.

A prominent algorithm in RL is the Q-Learning algorithm invented by Watkins [7]. The algorithm can be viewed as a stochastic-approximation counterpart of regular value iteration for discounted reward [3]. An attractive feature of this algorithm is its stability which is partly due to the fact that the iterates remain bounded. Also, the convergence analysis in Tsitsiklis [6] and Borkar and Meyn [2] requires proving boundedness of the iterates first. Two remarks are in order about these papers: (i) they prove boundedness in an *almost sure sense*, (ii) their proofs hold in an asymptotic sense in the number of iterations, k, and (iii) their proofs are quite sophisticated. We prove boundedness in an absolute sense; our result is independent of how the step size is decayed, and holds for any value of k. Also, our proof is relatively simple requiring only mathematical induction.

We have organized the remainder of this paper as follows. In Section 2, we describe the algorithm along with some notation needed thereafter. In Section 3, we prove our result. Section 4 concludes the paper.

2 Q-Learning

We first provide some background description and some notation that we will need. In an MDP, the goal is to find the optimal *action* in each state visited to maximize the value of a performance metric, e.g, long-run discounted reward. A *policy* defines the action in every state visited. Let $\mathcal{A}(i)$ denote the set of actions allowed and let $\mathcal{S}$ denote the set of states. We will assume that both $\mathcal{S}$ and $\mathcal{A}(i)$ for every $i \in \mathcal{S}$ are finite. Let $r(i,a,j)$ denote the immediate reward earned in going from state i to state j, when action a is selected in state i, and let λ denote the discounting factor (which could be $1/(1+R)$ where R denotes the rate of interest). Then with i as the starting state, for the policy π, the total discounted reward — generally referred to as discounted reward — is defined as:

$$\rho_i^\pi \equiv \mathsf{E}\left[\sum_{t=0}^{\infty}(\lambda)^t r\left(Z(t),\pi(Z(t)),Z(t+1)\right)|Z(0)=i\right],$$

where $Z(t)$ denotes the state of the system at time t, and $\pi(j)$ denotes the action selected in state j if policy π is pursued. If the Markov chain associated with the policy π is irreducible and aperiodic, the discounted reward, ρ_i^π, is independent of the starting state i.

In Q-Learning, which can be implemented in simulators or in real time [5], the algorithm iterates are the so-called Q-factors. $Q(i,a)$ will denote the Q-factor for state $i \in \mathcal{S}$ and $a \in \mathcal{A}(i)$. The updating in Q-Learning is asynchronous. One iteration of the algorithm is associated with each state transition in the simulator. In one transition, *only one* Q-factor is updated. Let $Q_k(i,a)$ denote the Q-factor for state i and action a in the kth iteration of the algorithm. When the system transitions from state i to state j and u denotes the action chosen in state i, then the Q-factor for state i and action u is updated as follows:

$$Q_{k+1}(i,u) = (1-\alpha(k))Q_k(i,u) + \alpha(k)\left[r(i,u,j) + \lambda \max_{b\in\mathcal{A}(j)} Q_k(j,b)\right]$$

where $\alpha(k)$, which denotes a step size in the kth iteration, must satisfy some standard stochastic-approximation conditions [4], and all other Q-factors are kept unchanged. Theoretically, the algorithm is allowed to run for infinitely many iterations until the Q-factors

converge. In the remainder of the paper, we will require that $\alpha(k) \leq 1$ for all k. Also, we will drop k from the notation, and simply refer to the step size as α.

3 Boundedness

We now present a result that will show boundedness of the Q-factors in Q-Learning.

Theorem 1 *In Q-Learning for discounted-reward MDPs, the iterate $Q_k(i,a)$ remains bounded for every state-action pair $(i,a) \in (\mathcal{S} \times \mathcal{A}(i))$ and for all k.*

Proof We first claim that for every $(i,a) \in (\mathcal{S} \times \mathcal{A}(i))$ and for any k in $\{1,2,\ldots\}$,

$$|Q_k(i,a)| \leq M(1+\lambda+\lambda^2+\cdots+\lambda^k), \tag{1}$$

where λ is the discounting factor and M is a positive finite number defined as follows:

$$M \equiv \max\{r_{\max}, \max_{(i,a)\in\mathcal{S},\mathcal{A}(i))} Q_0(i,a)\}, \tag{2}$$

where

$$r_{\max} \equiv \max_{i,j\in\mathcal{S},a\in\mathcal{A}(i)} |r(i,a,j)|. \tag{3}$$

Since we start with finite values for the Q-factors, $Q_0(i,a)$ is finite for every state-action pair $(i,a) \in (\mathcal{S} \times \mathcal{A}(i))$. Since the immediate rewards are finite by definition, $r_{\max}$ is also a finite quantity. Then, M too is finite from its definition in (2).

Boundedness follows from the claim in (1), since then if $k \to \infty$,

$$\limsup_{k\to\infty} |Q_k(i,a)| \leq M\frac{1}{1-\lambda}$$

for all $(i,a) \in (\mathcal{S} \times \mathcal{A}(i))$. Therefore, all we need to do to prove the result is to prove our claim in (1). We will use an induction argument.

In the kth iteration of the asynchronous algorithm, the update for (i, a) is either according to

Case 1:

$$Q_{k+1}(i,a) = (1-\alpha)Q_k(i,a) + \alpha\left[r(i,a,j) + \lambda \max_{b\in\mathcal{A}(j)} Q_k(j,b)\right],$$

or

Case 2:

$$Q_{k+1}(i,a) = Q_k(i,a).$$

Now, if the update is carried out as in Case 1:

$$\begin{aligned}
|Q_1(i,a)| &\leq (1-\alpha)|Q_0(i,a)| + \alpha|r(i,a,j) + \lambda \max_{b\in\mathcal{A}(j)} Q_0(j,b)| \\
&\leq (1-\alpha)M + \alpha M + \alpha\lambda M \text{ (from (3) and (2))} \\
&\leq (1-\alpha)M + \alpha M + \lambda M \text{ (from the fact that } \alpha \leq 1) \\
&= M(1+\lambda)
\end{aligned}$$

Now, if the update is carried out as in Case 2:

$$\begin{aligned}
|Q_1(i,a)| &= |Q_0(i,a)| \\
&\leq M < M(1+\lambda).
\end{aligned}$$

From the above, our claim in (1) is true for $k = 1$. Now assuming that the claim is true when $k = m$, we have that for all $(i, a) \in (\mathcal{S} \times \mathcal{A}(i))$.

$$|Q_m(i,a)| \leq M(1 + \lambda + \lambda^2 + \cdots + \lambda^m). \tag{4}$$

Now, if the update is carried out as in Case 1:

$$|Q_{m+1}(i,a)| \leq (1-\alpha)|Q_m(i,a)| + \alpha|r(i,a,j) + \lambda \max_{j\in\mathcal{A}(j)} Q_m(j,b)|$$

$$
\begin{aligned}
&\leq (1-\alpha)M(1+\lambda+\lambda^2+\cdots+\lambda^m) \\
&\quad +\alpha M+\alpha\lambda M(1+\lambda+\lambda^2+\cdots+\lambda^m) \text{ (from 4)} \\
&= M(1+\lambda+\lambda^2+\cdots+\lambda^m) \\
&\quad -\alpha M(1+\lambda+\lambda^2+\cdots+\lambda^m) \\
&\quad +\alpha M+\alpha\lambda M(1+\lambda+\lambda^2+\cdots+\lambda^m) \\
&= M(1+\lambda+\lambda^2+\cdots+\lambda^m) \\
&\quad -\alpha M(1+\lambda+\lambda^2+\cdots+\lambda^m) \\
&\quad +\alpha M+\alpha M(\lambda+\lambda^2+\cdots+\lambda^{m+1}) \\
&= M(1+\lambda+\lambda^2+\cdots+\lambda^m) \\
&\quad -\alpha M(1+\lambda+\lambda^2+\cdots+\lambda^m)+\alpha M \\
&\quad +\alpha M(\lambda+\lambda^2+\cdots+\lambda^m)+\alpha M\lambda^{m+1} \\
&= M(1+\lambda+\lambda^2+\cdots+\lambda^m)-\alpha M(1+\lambda+\lambda^2+\cdots+\lambda^m) \\
&\quad +\alpha M(1+\lambda+\lambda^2+\cdots+\lambda^m)+\alpha M\lambda^{m+1} \\
&= M(1+\lambda+\lambda^2+\cdots+\lambda^m)+\alpha M\lambda^{m+1} \\
&\leq M(1+\lambda+\lambda^2+\cdots+\lambda^m)+M\lambda^{m+1} \text{ (since } \alpha\leq 1) \\
&= M(1+\lambda+\lambda^2+\cdots+\lambda^m+\lambda^{m+1})
\end{aligned}
$$

Now, if the update is carried out as in Case 2:

$$
\begin{aligned}
|Q_{m+1}(i,a)| &= |Q_m(i,a)| \\
&\leq M(1+\lambda+\lambda^2+\cdots+\lambda^m) \\
&< M(1+\lambda+\lambda^2+\cdots+\lambda^m+\lambda^{m+1})
\end{aligned}
\tag{5}
$$

From the above, the claim in (1) is proved for $k=m+1$. ∎

4 Conclusions

Boundedness of iterates in Q-Learning (asynchronous) has been established in the literature with probability 1 in an asymptotic sense. We showed boundedness under weaker conditions — in particular that the result holds for any value of k (the number of iterations in the algorithm) independent of how the step size is decayed. Also, our proof is based on a simple induction argument. It is not clear if boundedness can be proved for average reward problems using a similar strategy, but a possible extension is to stochastic shortest path problems.

References

[1] D. Bertsekas and J. Tsitsiklis. *Neuro-dynamic programming.* Athena Scientific, MA, 1996.

[2] V. Borkar and S. Meyn. The ODE method for convergence of stochastic approximation and reinforcement learning. *SIAM Journal of Control and Optimization,* 38 (2): 447-469, 2000.

[3] M. Puterman. *Markov Decision Processes.* Wiley, NY, 1994.

[4] H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Statist.,* 22:400-407, 1951.

[5] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, MA, USA, 1998.

[6] J. Tsitsiklis. Asynchronous stochastic approximation and Q-Learning. *Machine Learning,* 16:185-202, 1994.

[7] C. Watkins. *Learning from Delayed Rewards.* Ph.D. thesis, Kings College, Cambridge, England, 1989.