Convergence Results for Single-Step On-Policy Reinforcement-Learning Algorithms

SATINDER SINGH baveja@cs.colorado.edu
Department of Computer Science
University of Colorado
Boulder, CO 80309-0430

TOMMI JAAKKOLA tommi@cse.ucsc.edu
Computer Science Department
University of California
Santa Cruz, CA 95064

MICHAEL L. LITTMAN mlittman@cs.duke.edu
Department of Computer Science
Duke University
Durham, NC 27708-0129

CSABA SZEPESVÁRI szepes@sol.cc.u-szeged.hu
Bolyai Institute of Mathematics
"Jozsef Attila" University of Szeged
Szeged 6720, Aradi vrt tere 1.
Hungary

Abstract. An important application of reinforcement learning (RL) is to finite-state control problems and one of the most difficult problems in learning for control is balancing the exploration/exploitation tradeoff. Existing theoretical results for RL give very little guidance on reasonable ways to perform exploration. In this paper, we examine the convergence of single-step on-policy RL algorithms for control. On-policy algorithms cannot separate exploration from learning and therefore must confront the exploration problem directly. We prove convergence results for several related on-policy algorithms with both decaying exploration and persistent exploration. We also provide examples of exploration strategies that can be followed during learning that result in convergence to both optimal values and optimal policies.

Keywords: reinforcement-learning, on-policy, convergence, Markov decision processes

1. Introduction

Most reinforcement-learning (RL) algorithms (Kaelbling et al., 1996; Sutton & Barto, 1997) for solving discrete optimal control problems use evaluation or *value* functions to cache the results of experience. This is useful because close approximations to optimal value functions lead directly to good control policies (Williams & Baird, 1993; Singh & Yee, 1994). Different RL algorithms combine new experience with old value functions to produce new and statistically improved value functions in different ways. All such algorithms face a tradeoff between exploitation and exploration (Thrun, 1992; Kumar & Varaiya, 1986; Dayan & Sejnowski, 1996), i.e., between choosing actions that are best according to the current state of knowledge, and actions that are not the current best but improve the state of knowledge and potentially yield higher payoffs in the future.

Following Sutton and Barto (1997), we distinguish between two types of RL algorithms: on-policy and off-policy. Off-policy algorithms may update estimated value functions on the basis of hypothetical actions, i.e., actions other than those actually executed—in this sense Q-learning (Watkins & Dayan, 1992) is an off-policy algorithm. On-policy algorithms, on the other hand, update value functions strictly on the basis of the experience gained from executing some (possibly non-stationary) policy. This distinction is important because off-policy algorithms can (at least conceptually) separate exploration from control while on-policy algorithms cannot. More precisely, in the case of on-policy algorithms, a convergence proof requires more details of the exploration to be specified than for off-policy algorithms, since the update rule depends a great deal on the actions taken by the system.

On-policy algorithms may prove to be important for several reasons. The analogue of the on-policy/off-policy distinction for RL prediction problems is the trajectory-based/trajectory-free distinction. Trajectory-based algorithms appear superior to trajectory-free algorithms for prediction when parameterized function approximators are used (Tsitsiklis & Van Roy, 1996). These results carry over empirically to the control case as well (Boyan & Moore, 1995; Sutton, 1996). In addition, multi-step prediction algorithms such as TD(λ)(Sutton, 1988) are more flexible and data efficient than single-step algorithms (TD(0)), and most natural multi-step algorithms for control are on-policy. These observations suggest that on-policy control algorithms are important and worthy of study.

In this paper, we examine the convergence of single-step (value updates based on the value of the "next" timestep only), on-policy RL algorithms for control. We do not address either function approximation or multi-step algorithms; this is the subject of our ongoing research. Earlier work has shown that there are off-policy RL algorithms that converge to optimal value functions (Watkins & Dayan, 1992; Dayan, 1992; Jaakkola et al., 1994; Tsitsiklis, 1994; Gullapalli & Barto, 1994; Littman & Szepesvári, 1996); we prove convergence results for several related on-policy algorithms. We also provide examples of policies that can be followed during learning that result in convergence to both optimal values and optimal policies. These results generalize naturally to off-policy algorithms, such as Q-learning, showing the convergence of many RL algorithms to optimal policies.

2. Solving Markov Decision Problems

Markov decision processes (MDPs) are widely used to model controlled dynamical systems in control theory, operations research and artificial intelligence (Puterman, 1994; Bertsekas, 1995; Barto et al., 1995). Let $S = 1, 2, \ldots, N$ denote the discrete set of states of the system, and let A be the discrete set of actions available to the system. The probability of making a transition from state s to state s' on action

a is denoted $P^a_{ss'}$ and the random payoff associated with that transition is denoted $r(s,a)$. A policy maps each state to a probability distribution over actions—this mapping can be invariant over time (stationary) or change as a function of the interaction history (non-stationary). For any policy π, we define a value function $V^\pi(s) = E_\pi\{\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s\}$, which is the expected value of the infinite-horizon sum of the discounted payoffs when the system is started in state s and the policy π is followed forever. Note that r_t and s_t are the payoff and state respectively at timestep t, and (r_t, s_t) is a (non-stationary) Markov process with transition probabilities given by the rules that r_t is distributed as $r(s_t, a_t)$ and the probability that $s_{t+1} = s$ is $P^{a_t}_{s_t s}$. Here, a_t is the action taken by the system at timestep t. The discount factor, $0 \leq \gamma < 1$, makes payoffs in the future less valuable than more immediate payoffs.

The solution of an MDP is an optimal policy π^* that simultaneously maximizes the value $V^{\pi^*}(s)$ of every state $s \in S$. It is known that a stationary deterministic optimal policy exists for every MDP (cf. Bertsekas' textbook, 1995). Hereafter, unless explicitly noted, all policies are assumed to be stationary. The value function associated with π^* is denoted V^*. Often it is convenient to associate values not with states but with state-action pairs, called Q values as in Watkins' (1989) Q-learning: $Q^\pi(s,a) = R(s,a) + \gamma E\{V^\pi(s')\}$, and $Q^*(s,a) = R(s,a) + \gamma E\{V^*(s')\}$, where s' is the random next state on executing action a in state s, and $R(s,a)$ is expected value of $r(s,a)$. Clearly, $\pi^*(s) = \operatorname{argmax}_a Q^*(s,a)$, and $V^*(s) = \max_a Q^*(s,a)$. The optimal Q values satisfy the recursive Bellman optimality equations (Bellman, 1957), $\forall s, a$:

$$Q^*(s,a) \;=\; R(s,a) + \gamma \sum_{s'} P^a_{ss'} \max_b Q^*(s',b). \tag{1}$$

In reinforcement learning, the quantities that define the MDP, P and R, are not known in advance. A RL algorithm must find an optimal policy by interacting with the MDP directly; because effective learning typically requires the algorithm to revisit every state many times, we assume the MDP is "communicating" (every state can be reached from every other state).

2.1. Off-Policy and On-Policy Algorithms

Most RL algorithms for solving MDPs are iterative, producing a sequence of estimates of either the optimal (Q-)value function or the optimal policy or both by repeatedly combining old estimates with the results of a new trial to produce new estimates.

A RL algorithm can be decomposed into two components. The *learning policy* is a non-stationary policy that maps experience (states visited, actions chosen, rewards received) into a current choice of action. The *update rule* is how the algorithm uses experience to change its estimate of the optimal value function.

In an off-policy algorithm, the update rule need not have any relation to the learning policy. Q-learning (Watkins, 1989) is an off-policy algorithm that estimates the optimal Q-value function as follows:

$$\begin{aligned} Q_{t+1}(s_t, a_t) &= (1 - \alpha_t(s_t, a_t))Q_t(s_t, a_t) \\ &\quad + \alpha_t(s_t, a_t)[r_t + \gamma \max_b (Q_t(s_{t+1}, b))], \end{aligned} \tag{2}$$

where Q_t is the estimate at the beginning of the t^{th} timestep, and s_t, a_t, r_t, and α_t are the state, action, reward, and step size (learning rate) at timestep t. This is an off-line algorithm as the update of $Q_t(s_t, a_t)$ depends on $\max_b(Q_t(s_{t+1}, b))$, which relies on comparing various "hypothetical" actions b. The convergence of the Q-learning algorithm does not put any strong requirements on the learning policy other than that every action is experienced in every state infinitely often. This can be accomplished, for example, using the random-walk learning policy, which chooses actions uniformly at random. Later, we describe several other learning policies that result in convergence when combined with the Q-learning update rule.

The update rule for SARSA(0)(Rummery, 1994; Rummery & Niranjan, 1994; John, 1994, 1995; Singh & Sutton, 1995; Sutton, 1996) is quite similar to Q-learning:

$$\begin{aligned} Q_{t+1}(s_t, a_t) &= (1 - \alpha_t(s_t, a_t))Q_t(s_t, a_t) \\ &\quad + \alpha_t(s_t, a_t)[r_t + \gamma Q_t(s_{t+1}, a_{t+1})]. \end{aligned} \tag{3}$$

The main difference is that Q-learning makes an update based on the greedy Q value of the successor state, s_{t+1}, while SARSA(0)[1] uses the Q value of the action a_{t+1} actually chosen by the learning policy. This makes SARSA(0) an on-policy algorithm, and therefore its conditions for convergence depend a great deal on the learning policy. In particular, because SARSA(0) learns the value of its own actions, the Q values can converge to optimality in the limit only if the learning policy chooses actions optimally in the limit. Section 3 provides some positive convergence results for two significant classes of learning policies.

Under a greedy learning policy (i.e., always select the action that is best according to the current estimate), the update rules for Q-learning and SARSA(0) are identical. The resulting RL algorithm would not converge to optimal solutions, in general, because the need for infinite exploration would not be satisfied. This helps illustrate the tension between adequate exploration and exploitation with regard to convergence to optimality.

2.2. Learning Policies

A learning policy selects an action at timestep t as a function of its experience history. In this paper, we consider several learning policies that make decisions based on a summary of history consisting of the current timestep t, the current state s, and the current estimate Q of the optimal Q-value function. Such a learning policy can be expressed as $\Pr(a|s, t, Q)$, the probability that action a is selected given the history.

We divide learning policies for MDPs into two broad categories; a *decaying exploration* learning policy becomes more and more like the greedy learning policy over time, a *persistent exploration* learning policy does not. The advantage of decaying exploration policies is that the actions taken by the system may converge to the optimal ones eventually, but with the price that their ability to adapt slows down. In contrast to this, persistent exploration learning policies can retain their adaptivity forever, but with the price that the actions of the system will not converge to optimality in the standard sense. We prove the convergence of SARSA(0) to optimal policies in the standard sense for a class of decaying exploration learning policies, and to optimal policies in a special sense defined below for a class of persistent exploration learning policies.

Consider the class of decaying exploration learning policies characterized by the following two properties:

1. each action is visited infinitely often in every state that is visited infinitely often,

2. in the limit, the learning policy is greedy with respect to the Q-value function with probability 1;

we label learning policies satisfying the above conditions as *GLIE*, which stands for "greedy in the limit with infinite exploration." An example of such a learning policy is certain forms of Boltzmann exploration:

$$\Pr(a|s,t,Q) = \frac{e^{\beta_t(s)Q(s,a)}}{\sum_{b\in A} e^{\beta_t(s)Q(s,b)}},$$

where $\beta_t(s)$ is the state-specific exploration coefficient for time t, which controls the rate of exploration in the learning policy. To meet condition 2 above, we would like β_t to be infinite in the limit, while to meet condition 1 above we would like β_t to not approach infinity too fast. In Appendix A, we show that $\beta_t(s) = \ln n_t(s)/C_t(s)$ satisfies the above requirements (where $n_t(s) \leq t$ is the number of times state s has been visited in the t timesteps, and $C_t(s)$ is defined in Appendix A). Another example of a GLIE learning policy is some forms of ϵ-greedy exploration (Sutton, 1996), which at timestep t in state s picks a random exploration action with probability $\epsilon_t(s)$ and the greedy action with probability $1-\epsilon_t(s)$. In Appendix A, we show that if $\epsilon_t(s) = c/n_t(s)$ for $0 < c < 1$, then ϵ-greedy exploration is GLIE.

We also analyze "restricted rank-based randomized learning policies" (*RRR*), a class of persistent exploration learning policies commonly used in practice. An RRR learning policy selects actions probabilistically according to the ranks of their Q values, choosing the greedy action with the highest probability and the action with the lowest Q value with the lowest probability. Different learning policies can be specified by different choices of the function $T : \{1,\ldots,m\} \rightarrow \Re$ that maps action ranks to probabilities. Here, m is the number of actions. For consistency, we require that $T(1) \geq T(2) \geq \cdots \geq T(m)$ and $\sum_{i=1}^{m} T(i) = 1$. At timestep t, the RRR learning policy chooses an action by first ranking the available actions according to the Q values assigned by the current Q-value function Q_t for the current state s_t. We use the notation $\rho(Q,s,a)$ to be the rank of action a in state s

based on $Q(s,\cdot)$ (e.g., if $\rho(Q,s,a)=1$ then $a=\mathrm{argmax}_b\, Q(s,b)$), with ties broken arbitrarily. Once the actions are ranked, the i^{th} ranking action is chosen with probability $T(i)$; that is, action a is chosen with probability $T(\rho(Q,s,a))$. The RRR learning policy is "restricted" in that it does not directly choose actions—it simply assigns probabilities to actions according to their ranks. Therefore, an RRR learning policy has the form $\Pr(a|s,t,Q)=T(\rho(Q_t,s,a))$.

To illustrate the use of the T function, we specify three well-known learning policies as RRR learning policies by the appropriate definition of T. The random-walk learning policy chooses action a in state s with probability $1/m$. To achieve this behavior with the RRR learning policy, simply define $T(i)=1/m$ for all i; actions will be equally likely regardless of their rank. The greedy learning policy can be specified by $T(1)=1$, $T(i)=0$ when $1<i\le m$; it deterministically selects the action with the highest Q value. Similarly, ϵ-greedy exploration can be specified by defining $T(1)=1-\epsilon+\epsilon/m$, $T(i)=\epsilon/m$, $1<i\le m$. This policy takes the greedy action with probability $1-\epsilon$ and a random action otherwise. To satisfy the condition that $T(1)\ge T(2)\ge\cdots\ge T(m)$, we require that $0\le\epsilon\le 1$.

Another commonly used persistent exploration learning policy is Boltzmann exploration with a fixed exploration parameter. Note there is no choice of T that specifies Boltzmann exploration; Boltzmann exploration is not an RRR learning policy as the probability of choosing an action depends on the actual Q values and not only on the ranks of actions in $Q(\cdot)$.

3. Results

Below we prove results on the convergence of SARSA(0) under the two separate cases of GLIE and RRR learning policies.

3.1. Convergence of SARSA(0) under GLIE Learning Policies

To ensure the convergence of SARSA(0), we require a lookup-table representation for the Q values and infinite visits to every state-action pair, just as for Q-learning. Unlike Q-learning, however, SARSA(0) is an on-policy algorithm and, in order to achieve its convergence to optimality, we have to further assume that the learning policy becomes greedy in the limit.

To state these assumptions and the resulting convergence more formally, we note first that due to the dependence on the learning policy, SARSA(0) does not directly fall under the previously published convergence theorems (Dayan & Sejnowski, 1994; Jaakkola et al., 1994; Tsitsiklis, 1994; Szepesvári & Littman, 1996). Only a slight extension is needed, however, and this is presented in the form of Lemma 1 below (extending Theorem 1 of Jaakkola et al., 1994, and Lemma 12 of Szepesvári & Littman, 1996). For clarity, we will not present the lemma in full generality.

LEMMA 1 *A random iterative process*

$$\Delta_{t+1}(x) = (1 - \alpha_t(x))\Delta_t(x) + \alpha_t(x)F_t(x),\ x \in X,\ t = 0, 1, 2, \ldots$$

converges to zero with probability one (w.p.1) if the following properties hold:

1. *the set of possible states X is finite.*
2. $0 \leq \alpha_t(x) \leq 1$, $\sum_t \alpha_t(x) = \infty$, $\sum_t \alpha_t^2(x) < \infty$ *w.p.1, where the probability is over the learning rates α_t.*
3. $||E\{F_t(\cdot)|P_t\}||_W \leq \kappa||\Delta_t||_W + c_t$, *where $\kappa \in [0, 1)$ and c_t converges to zero w.p.1.*
4. $\mathrm{Var}\{F_t(x)|P_t\} \leq K(1 + ||\Delta_t||_W)^2$, *where K is some constant.*

Here P_t is an increasing sequence of σ-fields that includes the past of the process. In particular, we assume that $\alpha_t, \Delta_t, F_{t-1} \in P_t$. The notation $||\cdot||_W$ refers to some (fixed) weighted maximum norm.

Let us first clarify how this lemma relates to the learning algorithms that are the focus of this paper. The sequence of visited states s_t and selected actions a_t are captured by defining the learning rates α_t in the following way. We can define $x_t = (s_t, a_t)$ and further require that $\alpha_t(x) = 0$ whenever $x \neq x_t$. With these definitions, the iterative process reduces to

$$\Delta_{t+1}(s_t, a_t) = (1 - \alpha_t(s_t, a_t))\Delta_t(s_t, a_t) + \alpha_t(s_t, a_t)F_t(s_t, a_t),$$

which resembles more closely the updates of the on-line algorithms such as SARSA(0) (Equation 3). Also, note that the lemma shows the convergence of Δ to zero rather than to some non-zero optimal values. The intended meaning of Δ is $Q_t - Q^*$, i.e., the difference between the current Q values, Q_t, and the target Q values, Q^*, that are attained asymptotically.

The extension provided by our formulation of the lemma is the fact that the contraction property (the third condition) need not be strict; strict contraction is now required to hold only asymptotically. This relaxation makes the theorem more widely applicable.

Proof: While we have stated that the lemma extends previous results such as the Theorem 1 of Jaakkola et al. (1994) and Lemma 12 of Szepesvári & Littman (1996), the proof of our lemma is, however, already almost fully contained in the proofs of these results (requiring only minor, largely notational changes). We, thus, refrain from repeating that proof here. ■

We can now use Lemma 1 to show the convergence of SARSA(0).

THEOREM 1 *In finite state-action* MDP*s, the Q_t values computed by the* SARSA(0) *rule (see Equation 3) converge to Q^* and the learning policy π_t converges to an optimal policy π^* if the learning policy is GLIE, the conditions on the immediate rewards and state transitions listed in Section 2 hold and if the following additional conditions are satisfied:*

1. *The Q values are stored in a lookup table.*
2. *The learning rates satisfy* $0 \le \alpha_t(s,a) \le 1$, $\sum_t \alpha_t(s,a) = \infty$ *and* $\sum_t \alpha_t^2(s,a) < \infty$ *and* $\alpha_t(s,a) = 0$ *unless* $(s,a) = (s_t, a_t)$.
3. $Var\{r(s,a)\} < \infty$.

Proof: The correspondence to Lemma 1 follows from associating X with the set of state-action pairs (s,a), $\alpha_t(x)$ with $\alpha_t(s,a)$ and $\Delta_t(s,a)$ with $Q_t(s,a) - Q^*(s,a)$. It follows that

$$\Delta_{t+1}(s_t,a_t) = (1-\alpha_t(s_t,a_t))\Delta_t(s_t,a_t) + \alpha_t(s_t,a_t)F_t(s_t,a_t),$$

where

$$\begin{aligned} F_t(s_t,a_t) &= r_t + \gamma \max_{b\in A} Q_t(s_{t+1},b) - Q^*(s_t,a_t) \\ &\quad + \gamma[Q_t(s_{t+1},a_{t+1}) - \max_{b\in A} Q_t(s_{t+1},b)] \\ &\stackrel{def}{=} r_t + \gamma \max_{b\in A} Q_t(s_{t+1},b) - Q^*(s_t,a_t) + C_t(Q) \\ &\stackrel{def}{=} F_t^Q(s_t,a_t) + C_t(s_t,a_t), \end{aligned}$$

where F_t^Q would be the corresponding F_t in Lemma 1 if the algorithm under consideration were Q-learning. Further, we define $F_t(s,a) = C_t(s,a) = 0$ if $(s,a) \ne (s_t,a_t)$ and denote the σ-field generated by the random variables $\{s_t, \alpha_t, a_t, r_{t-1}, \ldots, s_1, \alpha_1, a_1, Q_0\}$ by P_t. Note that $Q_t, Q_{t-1}, \ldots, Q_0$ are P_t-measurable and, thus, both Δ_t and F_{t-1} are P_t-measurable, satisfying the measurability conditions of Lemma 1.

It is well-known that for Q-learning $||E\{F_t^Q(\cdot,\cdot)\,|\,P_t\}|| \le \gamma||\Delta_t||$ for all t, where $||\cdot||$ is the maximum norm. In other words, the expected update operator is a contraction mapping. The only difference between the current F_t and F_t^Q for Q-learning is the presence of C_t. Therefore,

$$\begin{aligned} ||E\{F_t(\cdot,\cdot)|P_t\}|| &\le ||E\{F_t^Q(\cdot,\cdot)\,|\,P_t\}|| + E\{C_t(\cdot,\cdot)\,|\,P_t\}|| && (4) \\ &\le \gamma||\Delta_t|| + ||E\{C_t(\cdot,\cdot)\,|\,P_t\}||. && (5) \end{aligned}$$

Identifying $c_t = ||E\{C_t(\cdot,\cdot)\,|\,P_t\}||$ in Lemma 1, we are left with showing that c_t converges to zero w.p.1. This, however, follows (1) from our assumption of a GLIE policy (i.e., that non-greedy actions are chosen with vanishing probabilities), (2) the assumption of finiteness of the MDP, and (3) the fact that $Q_t(s,a)$ stays bounded during learning. To verify the boundedness property, we note that the SARSA(0) Q values can be *upper* bounded by the Q values of a Q-learning process that updates exactly the same state-action pairs in the same order as the SARSA(0) process. Similarly, the SARSA(0) Q values are *lower* bounded by the Q values of a Q-learning process that uses a min instead of a max in the update rule (cf. Equation 2) and updates exactly the same state-action pairs in the same order as the SARSA(0) process. Both the lower-bounding and the upper-bounding Q-learning processes are convergent and have bounded Q values.

The condition on the variance of F_t follows from the similar property of F_t^Q. ∎

Note that if a GLIE learning policy is used with the Q-learning update rule, one gets convergence to both the optimal Q-value function and an optimal policy. This begins to address a significant outstanding question in the theory of reinforcement learning: How do you a learn a policy that achieves high reward in the limit *and* during learning? Previous convergence results for Q-learning guarantee that the optimal Q-value function is reached in the limit; this is important because the longer the learning process goes on, the closer to optimal the greedy policy with respect to the learned Q-value function will be. However, this provides no useful guidance for selecting actions during learning. Our results, in contrast, show that it is possible to follow a policy *during learning* that approaches optimality over time.

The properties of GLIE policies imply that for any RL algorithm that converges to the optimal value function and whose estimates stay bounded (e.g., Q-learning, and ARTDP of Barto et al., 1995), using GLIE learning policies will ensure a concurrent convergence to an optimal policy. However, to get an implementable RL algorithm, one still has to specify a suitable learning policy that guarantees that every action is attempted in every state infinitely often (i.e., $\sum_t \alpha_t(s,a) = \infty$). In the Appendix, we prove that, if the probability of choosing any particular action in any given state sums up to infinity, then the above condition is indeed satisfied. To illustrate this, we derive two learning strategies that are GLIE.

3.2. Convergence of SARSA(0) under RRR Learning Policies

This section proves two separate results concerning a class of persistent exploration learning policies: (1) the SARSA(0) update rule combined with an RRR learning policy converges to a well-defined Q-value function and policy, and (2) the resulting policy is optimal, in a sense we will define.

As mentioned earlier, an RRR learning policy chooses actions probabilistically by their ranking according to the current Q-value function; a specific learning policy is specified by the function T, a probability distribution over action ranks. A *restricted policy* $\bar{\pi} : S \rightarrow \Pi(A, \{1, \ldots, m\})$ ranks actions in each state (recall that m denotes the number of actions), i.e., $\bar{\pi}(s)$ is a bijection between A and $\{1, \ldots, m\}$. For convenience, we use the notation $\bar{\pi}(s,a)$ to denote the assigned rank of action a in state s, i.e., to denote $\bar{\pi}(s)(a)$. The mapping $\bar{\pi}$ represents a policy in the sense that an agent following restricted policy $\bar{\pi}$ from state s chooses action a with probability $T(\bar{\pi}(s,a))$, the probability of the rank assigned by $\bar{\pi}$ to action a in state s.

Consider what happens when the SARSA(0) update rule is used to learn the value of a fixed restricted policy $\bar{\pi}$. Standard convergence results for Q-learning can easily be used to show that the Q_t values will converge to the Q-value function of $\bar{\pi}$. Specifically, Q_t will converge to $Q^{\bar{\pi}}$, defined as the unique solution to

$$Q^{\bar{\pi}}(s,a) = R(s,a) + \gamma \sum_{s' \in S} P^a_{ss'} \sum_{a' \in A} T(\bar{\pi}(s',a'))Q^{\bar{\pi}}(s',a'),\ (s,a) \in S \times A. \quad (6)$$

When an RRR learning policy is followed, the situation becomes a bit more complex. Upon entering state s, the probability that the learning policy will choose, for example, the rank 1 action is fixed at $T(1)$; however, the identity of that action changes as a function of the current Q-value function estimate $Q_t(\cdot,\cdot)$. The natural extension of Equation 6 to an RRR learning policy would be for the target of convergence of Q_t in SARSA(0) to be

$$\bar{Q}(s,a) = R(s,a) + \gamma \sum_{s' \in S} P^a_{ss'} \sum_{a' \in A} T(\rho(\bar{Q},s',a'))\bar{Q}(s',a'),\ (s,a) \in S \times A. \quad (7)$$

Recall that $\rho(\bar{Q},s',a')$ represents the rank of action a' according to the Q values $\bar{Q}$ of state s'. The only change between Equation 6 and Equation 7 is that the latter uses an assignment of ranks that is based upon the recursively defined Q-value function $\bar{Q}$, whereas the former uses a fixed assignment of ranks. Using the theory of generalized MDPs(Szepesvári & Littman, 1996), we can show that this difference is not important from the perspective of proving the existence and uniqueness of the solution of Equation 7.

Define

$$\bigotimes_a Q(s,a) = \sum_{a \in A} T(\rho(Q,s,a))Q(s,a); \quad (8)$$

now Equation 7 can be rewritten

$$\bar{Q}(s,a) = R(s,a) + \gamma \sum_{s' \in S} P^a_{ss'} \bigotimes_{a'} \bar{Q}(s',a'),\ (s,a) \in S \times A. \quad (9)$$

As long as $\bigotimes$ satisfies the non-expansion property that

$$\left| \bigotimes_a Q(s,a) - \bigotimes_a Q'(s,a) \right| \leq \max_a |Q(s,a) - Q'(s,a)|$$

for all Q-value functions Q and Q' and all states s, then Equation 9 has a solution and it is unique (Szepesvári & Littman, 1996). The non-expansion property of $\bigotimes$ can be verified by the following argument.

- Consider a family of operators $\bigotimes^i_a Q(s,a) = i$th largest value of $Q(s,a)$ for each $1 \leq i \leq m$. These are all non-expansions.
- Define $\bigotimes'_a Q(s,a) = \sum_i T(i) \bigotimes^i_a Q(s,a)$; it is a non-expansion as long as every $\bigotimes^i_a$ is and T is a fixed probability distribution.
- It is clear that $\bigotimes'_a Q(s,a) = \bigotimes_a Q(s,a)$ as defined in Equation 8, so $\bigotimes$ is a non-expansion also.

Therefore, $\bar{Q}$ exists and is unique. We next show that $\bar{Q}$ is, in fact, the target of convergence for SARSA(0).

THEOREM 2 *In finite state-action* MDP*s, the Q_t values computed by the* SARSA(0) *rule (see Equation 3) converge to $\bar{Q}$ and the learning policy π_t converges to a restricted optimal policy $\bar{\pi}$ if the learning policy is RRR, the conditions on the immediate rewards and state transitions listed in Section 2 hold, and if the following additional conditions are satisfied:*

1. $\Pr(a_{t+1} = a \mid Q_t, s_{t+1}) = T(\rho(Q_t, s_{t+1}, a_{t+1}))$.
2. *The Q values are stored in a lookup table.*
3. *The learning rates satisfy* $0 \leq \alpha_t(s,a) \leq 1$, $\sum_t \alpha_t(s,a) = \infty$, $\sum_t \alpha_t^2(s,a) < \infty$, *and* $\alpha_t(s,a) = 0$ *unless* $(s,a) = (s_t, a_t)$.
4. $Var\{r(s,a)\} < \infty$.

Proof: The result readily follows from Lemma 1 (or Theorem 1 Jaakkola et al., 1994) and the proof follows nearly identical lines as that of Theorem 3.1. First, we associate X (of Lemma 1) with the set of state-action pairs (s,a), $\alpha_t(x)$ with $\alpha_t(s,a)$, but here we set $\Delta_t(s,a) = Q_t(s,a) - \bar{Q}(s,a)$. Again, it follows that

$$\Delta_{t+1}(s_t, a_t) = (1 - \alpha_t(s_t, a_t))\Delta_t(s_t, a_t) + \alpha_t(s_t, a_t) F_t(s_t, a_t),$$

where now

$$F_t(s_t, a_t) = r_t + \gamma Q_t(s_{t+1}, a_{t+1}) - \bar{Q}(s_t, a_t).$$

Further, we define $F_t(s,a) = C_t(s,a) = 0$ if $(s,a) \neq (s_t, a_t)$ and denote the σ-field generated by the random variables $\{s_t, \alpha_t, a_t, r_{t-1}, \ldots, s_1, \alpha_1, a_1, Q_0\}$ by P_t. Note that $Q_t, Q_{t-1}, \ldots, Q_0$ are P_t-measurable and, thus, both Δ_t and F_{t-1} are P_t-measurable, satisfying the measurability conditions of Lemma 1.

Substituting the right-hand side of Equation 7 for $\bar{Q}(s_t, a_t)$ in the definition of F_t together with the properties of sampling r_t, s_{t+1} and a_{t+1} yields that

$$\begin{aligned} E\{F_t(s_t, a_t) \mid P_t\} &= \gamma \Big(E\{Q_t(s_{t+1}, a_{t+1}) \mid P_t\} - \\ &\qquad \sum_{s' \in S} P^{a_t}_{s_t s'} \sum_{a' \in A} T(\rho(\bar{Q}, s', a')) \bar{Q}(s', a') \Big) \\ &= \gamma \Big(\sum_{s' \in S} P^{a_t}_{s_t s'} \sum_{a' \in A} T(\rho(Q_t, s', a')) Q_t(s', a') - \\ &\qquad \sum_{s' \in S} P^{a_t}_{s_t s'} \sum_{a' \in A} T(\rho(\bar{Q}, s', a')) \bar{Q}(s', a') \Big) \\ &\leq \gamma \|Q_t - \bar{Q}\| \\ &= \gamma \|\Delta_t\|, \end{aligned}$$

where in the first equation we have exploited the fact that $E\{r_t|s_t, a_t\} = R(s_t, a_t)$, in the second equation that $\Pr(s_{t+1} \mid s_t, a_t) = P^{a_t}_{s_t s_{t+1}}$ and that $\Pr(a_{t+1} = a \mid Q_t, s_{t+1}) = T(\rho(Q_t, s_{t+1}, a))$ (condition 1), whereas the inequality comes from the properties of rank-based averaging (see Lemma 7 and Theorems 9 and 10 of Szepesvári & Littman's (1996) technical report. Finally, it is not hard to prove that the variance of F_t given the past P_t satisfies condition 4 and, therefore, we do not include it here.

■

We have shown that SARSA(0) with an RRR learning policy converges to $\bar{Q}$. Next, we show that $\bar{Q}$ is, in a sense, an optimal Q-value function.

An *optimal restricted policy* is one that has the highest expected total discounted reward of all restricted policies. The *greedy restricted policy* for a Q-value function Q is $\bar{\pi}(s, a) = \rho(Q, s, a)$; it assigns each action the rank of its corresponding Q value. Note that this is the policy dictated by the RRR learning policy for a fixed Q-value function Q.

The greedy restricted policy for Q^* (the optimal Q-value function of the MDP) is not an optimal restricted policy in general, so the Q-learning rule in Equation 2 does not find an optimal restricted policy. However, the next theorem shows that the greedy restricted policy for $\bar{Q}$ (Equation 7) is an optimal restricted policy. This $\bar{Q}$ function is very similar to Q^*, except that actions are weighted according to the greedy restricted policy instead of the standard greedy policy.

THEOREM 3 *The greedy restricted policy with respect to $\bar{Q}$ is an optimal restricted policy.*

Proof: We construct an alternate MDP so that every restricted policy in the original MDP is in one-to-one correspondence with (and has the same value as) a deterministic stationary policy in the alternate MDP. Note that, as a result of the equivalence of value functions, the optimal policy of the alternate MDP will correspond to an optimal restricted policy of the original MDP (the restricted policy that achieves the best values for each of the states) and, thus, the theorem will follow if we show that the optimal policy in the alternate MDP corresponds to the greedy restricted policy with respect to $\bar{Q}$.

The alternate MDP is defined by $\langle S, \bar{A}, \bar{R}, \bar{P}, \gamma \rangle$. Its action space is $\bar{A} = \Pi(A, \{1, \ldots, m\})$, i.e., it is the set of all bijections from A to $\{1, \ldots, m\}$. The rewards are

$$\bar{R}(s, \mu) = \sum_{a \in A} T(\mu(a)) R(s, a),$$

and the transition probabilities are given by $\bar{P}^{\mu}_{ss'} = \sum_{a \in A} T(\mu(a)) P^{a}_{ss'}$. Here, μ is an element of $\bar{A}$. One can readily check that the value of a restricted policy $\bar{\pi}$ is just the value of $\bar{\pi}$ in the alternate MDP.

The value of the greedy restricted policy with respect to $\bar{Q}$ in the original MDP is

$$\bar{V}(s) = \sum_{a \in A} T(\rho(\bar{Q}, s, a)) \bar{Q}(s, a). \tag{10}$$

Substituting the definition of $\bar{Q}$ from Equation 7 into Equation 10 results in

$$\bar{V}(s) = \sum_{a \in A} T(\rho(\bar{Q}, s, a)) \left(R(s,a) + \gamma \sum_{s' \in S} P^a_{ss'} \sum_{a' \in A} T(\rho(\bar{Q}, s', a'))\bar{Q}(s', a') \right).$$

Using Equation 10 once again, we find that $\bar{V}$ satisfies the recurrence equation

$$\bar{V}(s) = \sum_{a \in A} T(\rho(\bar{Q}, s, a)) \left(R(s,a) + \gamma \sum_{s' \in S} P^a_{ss'} \bar{V}(s') \right). \tag{11}$$

Meanwhile, the optimum value of the alternate MDP satisfies

$$\begin{aligned} \bar{V}^*(s) &= \max_{\mu \in \bar{A}} \left(\bar{R}(s,\mu) + \gamma \sum_{s' \in S} \bar{P}^\mu_{ss'} \bar{V}^*(s') \right) \\ &= \max_{\mu \in \bar{A}} \left(\sum_{a \in A} T(\mu(a)) R(s,a) + \gamma \sum_{s' \in S} \left(\sum_{a \in A} T(\mu(a)) P^a_{ss'} \right) \bar{V}^*(s') \right) \\ &= \max_{\mu \in \bar{A}} \sum_{a \in A} T(\mu(a)) \left(R(s,a) + \gamma \sum_{s' \in S} P^a_{ss'} \bar{V}^*(s') \right). \end{aligned} \tag{12}$$

The highest value permutation is the one that assigns the highest probabilities to the actions with the highest Q values and the lowest probabilities to the actions with the lowest Q values. Therefore, the recurrence in Equation 12 is the same as that in Equation 11, so, by uniqueness, $\bar{V}^* = \bar{V}$. This means the greedy restricted policy with respect to $\bar{Q}$ is the optimal restricted policy. ■

As a corollary of Theorem 3.2, given a communicating MDP and a RL algorithm that follows an RRR learning policy specified by T where $T(i) > 0$ for all $1 \leq i \leq m$, SARSA(0) converges to an optimal restricted policy.

We conjecture that the same result does not hold for persistent Boltzmann exploration because related synchronous algorithms do not have a unique target of convergence (Littman, 1996).

4. Conclusion

In this paper, we have provided convergence results for SARSA(0) under two different learning policy classes; one ensures optimal behavior in the limit and the other ensures behavior optimal with respect to constraints imposed by the exploration strategy. To the best of our knowledge, these constitute the first convergence results for any on-policy algorithm. However, these are very basic results because they apply only to the lookup-table case, and more importantly because they do not seem to extend naturally to general multi-step on-policy algorithms.

Acknowledgments

We thank Rich Sutton for help and encouragement.

Appendix A

GLIE Learning Policies

Here, we present conditions on the exploration parameter in the commonly used Boltzmann exploration and ϵ-greedy exploration strategies to ensure that both infinite exploration and greedy in the limit conditions are satisfied.

In a communicating MDP, every state gets visited infinitely often as long as each action is chosen infinitely often in each state (this is a consequence of the Borel-Cantelli Lemma (Breiman, 1992); all we have to ensure is that in each state each action gets chosen infinitely often in the limit. Consider some state s. Let $t_s(i)$ represent the timestep at which the i^{th} visit to state s occurs. Consider some action a. The probability with which action a is executed at the i^{th} visit to state s is denoted $\Pr(a|s, t_s(i))$ (i.e, $\Pr(a = a_t|s_t = s, t_s(i) = t)$).

We would like to show that if the sum of the probabilities with which action a is chosen is infinite, i.e., $\sum_{i=1}^{\infty} \Pr(a|s, t_s(i)) = \infty$, then the number of times action a gets executed in state s is infinite w.p.1. This would follow directly from the Borel-Cantelli Lemma if the probabilities of selecting action a at the different i were independent. Unfortunately, in this case the random choice of action at the i^{th} visit to state s affects the probabilities at the $i + 1^{st}$ visit to state s (through the evolution of the Q-value function). However, if there exists another stochastic process that also sums to infinity, lower bounds the sequence of probabilities of selecting action a, and satisfies the independence conditions required by the Borel-Cantelli Lemma, then again the result would follow. We state this below more formally.

OBSERVATION 1 *Consider a stochastic process* $\{p_i\}_{i=1}^{\infty}$ *with* $0 \leq p_i \leq 1$ *for all* i. *Let random variable* S_i *be* 1 *with probability* p_i *and* 0 *with probability* $1 - p_i$. *Further, let* $N(n) = \sum_{i=1}^{n} S_i$. *If there exists another stochastic process* $\{c_i\}_{i=1}^{\infty}$ *such that* $0 \leq c_i \leq p_i$ *for all* i, $\sum_{i=1}^{\infty} c_i = \infty$, *and all finite subsets of* $\{c_i\}$ *are independent, then* $\lim_{n \to \infty} N(n) = \infty$ *w.p.1.*

Proof: Let $\hat{S}_i$ be equal to 1 with probability c_i and 0 with probability $1 - c_i$, then the Borel-Cantelli Lemma proves that $\lim_{n \to \infty} \hat{N}(n) = \infty$ w.p.1., where $\hat{N}(n) = \sum_{i=1}^{n} \hat{S}_i$. However, since $p_i \geq c_i$, the result must also follow for $N(n)$. ∎

A.1. Boltzmann Exploration

In Boltzmann exploration,

$$\Pr(a|s,t,Q) = \frac{e^{\beta_t(s)Q(s,a)}}{\sum_{b\in A} e^{\beta_t(s)Q(s,b)}},$$

where $\beta_t(s)$ is the state-specific exploration coefficient for time t. Let the number of visits to state s in timestep t be denoted as $n_t(s)$ and assume that $r(s,a)$ has a finite range. We know that $\sum_{i=1}^{\infty} c/i = \infty$; therefore, to meet the conditions of Observation A, we will ensure that for all actions $a \in A$, $\Pr(a|s,t_s(i)) \geq c/i$ (with $c \leq 1$). To do that we need for all a:

$$\begin{aligned}
\frac{e^{\beta_t(s)Q_t(s,a)}}{\sum_{b\in A} e^{\beta_t(s)Q_t(s,b)}} &\geq \frac{c}{n_t(s)} \\
n_t(s)e^{\beta_t(s)Q_t(s,a)} &\geq c\sum_{b\in A} e^{\beta_t(s)Q_t(s,b)} \\
n_t(s)e^{\beta_t(s)Q_t(s,a)} &\geq cme^{\beta_t(s)Q_t(s,b_{\max})} \\
\frac{n_t(s)}{cm} &\geq e^{\beta_t(s)(Q_t(s,b_{\max})-Q_t(s,a))} \\
\ln n_t(s) - \ln cm &\geq \beta_t(s)(Q_t(s,b_{\max}) - Q_t(s,a)),
\end{aligned}$$

where $b_{\max} = \text{argmax}_{b\in A}\, Q_t(s,b)$ above and m is the number of actions. Further, let $c = 1/m$. Taken together, this means that we want $\beta_t(s) \leq \ln n_t(s)/C_t(s)$ where $C_t(s) = \max_a |Q_t(s,b_{\max}) - Q_t(s,a)|$. Note that $C_t(s)$ is bounded because the Q values remain bounded.

It should also be clear that for every s, $\lim_{t\to\infty} n_t(s) = \infty$, and therefore

$$\lim_{t\to\infty} \beta_t(s) \leq \lim_{t\to\infty} \frac{\ln n_t(s)}{C_t(s)} = \infty;$$

this means that Boltzmann exploration with $\beta_t(s) = \ln n_t(s)/C_t(s)$ will be greedy in the limit.

A.2. ϵ-Greedy Exploration

In ϵ-greedy exploration we pick a random exploration action with probability $\epsilon_t(s)$ and the greedy action with probability $1-\epsilon_t(s)$. Let $\epsilon_t(s) = c/n_t(s)$ with $0 < c < 1$. Then, $\Pr(a|s,t_s(i)) \geq \epsilon_t(s)/m$, where m is the number of actions. Therefore, Observation A combined with the fact that $\sum_{i=1}^{\infty} c/i = \infty$ implies that for all s, $\sum_{i=1}^{\infty} \Pr(a|s,t_s(i)) = \infty$. Further, for all s, $\lim_{t\to\infty} n_t(s) = \infty$, and, therefore, $\lim_{t\to\infty} \epsilon_t(s) = 0$, ensuring that the learning policy is greedy in the limit. Therefore, if $\epsilon_t(s) = c/n_t(s)$ then ϵ-greedy exploration is GLIE for $0 < c < 1$.

Notes

1. The name is a reference to the fact that it is a single-step algorithm that makes updates on the basis of a State, Action, Reward, State, Action 5-tuple.

References

Andrew G. Barto, S. J. Bradtke, and Satinder Singh (1995). Learning to act using real-time dynamic programming. *Artificial Intelligence*, 72(1):81–138.

Richard Bellman (1957). *Dynamic Programming.* Princeton University Press, Princeton, NJ.

Dimitri P. Bertsekas (1995). *Dynamic Programming and Optimal Control.* Athena Scientific, Belmont, Massachusetts. Volumes 1 and 2.

Justin A. Boyan and Andrew W. Moore.(1995). Generalization in reinforcement learning: Safely approximating the value function. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 369–376, Cambridge, MA. The MIT Press.

Leo Breiman.(1992). *Probability.* Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania.

Peter Dayan (1992). The convergence of TD(λ) for general λ. *Machine Learning*, 8(3):341–362.

Peter Dayan and Terrence J. Sejnowski (1994). TD(λ) converges with probability 1. *Machine Learning*, 14(3).

Peter Dayan and Terrence J. Sejnowski (1996). Exploration bonuses and dual control. *Machine Learning*, 25:5–22.

Vijaykumar Gullapalli and Andrew G. Barto (1994). Convergence of indirect adaptive asynchronous value iteration algorithms. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 695–702, San Mateo, CA. Morgan Kaufmann.

Tommi Jaakkola, Michael I. Jordan, and Satinder Singh (1994). On the convergence of stochastic iterative dynamic programming algorithms. *Neural Computation*, 6(6):1185–1201, November.

George H. John (1994). When the best move isn't optimal: Q-learning with exploration. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, page 1464, Seattle, WA.

George H. John (1995). When the best move isn't optimal: Q-learning with exploration. Unpublished manuscript, available through URL `ftp://starry.stanford.edu/pub/gjohn/papers/rein-nips.ps`.

Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285.

P. R. Kumar and P. P. Varaiya (1986). *Stochastic Systems: Estimation, Identification, and Adaptive Control.* Prentice Hall, Englewood Cliffs, NJ.

Michael L. Littman and Csaba Szepesvári (1996). A generalized reinforcement-learning model: Convergence and applications. In Lorenza Saitta, editor, *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 310–318.

Michael Lederman Littman (1996). *Algorithms for Sequential Decision Making.* PhD thesis, Department of Computer Science, Brown University, February. Also Technical Report CS-96-09.

Martin L. Puterman (1994). *Markov Decision Processes—Discrete Stochastic Dynamic Programming.* John Wiley & Sons, Inc., New York, NY.

G. A. Rummery (1994). *Problem solving with reinforcement learning.* PhD thesis, Cambridge University Engineering Department.

G. A. Rummery and M. Niranjan (1994). On-line Q-learning using connectionist systems. Technical Report CUED/F-INFENG/TR 166, Cambridge University Engineering Department.

Satinder P. Singh and Richard S. Sutton (1996). Reinforcement learning with replacing eligibility traces. *Machine Learning*, 22(1/2/3):123–158.

Satinder Pal Singh and Richard C. Yee (1994). An upper bound on the loss from approximate optimal-value functions. *Machine Learning*, 16:227.

Rich Sutton and Andy Barto (1997). *An Introduction to Reinforcement Learning.* The MIT Press, forthcoming.

Richard S. Sutton (1988). Learning to predict by the method of temporal differences. *Machine Learning*, 3(1):9–44.

Richard S. Sutton (1996). Generalization in reinforcement learning: Successful examples using sparse coarse coding. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, Cambridge, MA. The MIT Press.

Csaba Szepesvári and Michael L. Littman (1996). Generalized Markov decision processes: Dynamic-programming and reinforcement-learning algorithms. Technical Report CS-96-11, Brown University, Providence, RI.

Sebastian B. Thrun (1992). The role of exploration in learning control. In David A. White and Donald A. Sofge, editors, *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches.* Van Nostrand Reinhold, New York, NY.

John N. Tsitsiklis (1994). Asynchronous stochastic approximation and Q-learning. *Machine Learning*, 16(3):185–202, September 1994.

John N. Tsitsiklis and Benjamin Van Roy (1996). An analysis of temporal-difference learning with function approximation. Technical Report LIDS-P-2322, Massachusetts Institute of Technology, March. Available through URL `http://web.mit.edu/bvr/www/td.ps`. To appear in *IEEE Transactions on Automatic Control.*

Christopher J. C. H. Watkins (1989). *Learning from Delayed Rewards.* PhD thesis, King's College, Cambridge, UK.

Christopher J. C. H. Watkins and Peter Dayan (1992). Q-learning. *Machine Learning*, 8(3):279–292.

Ronald J. Williams and Leemon C. Baird, III (1993). Tight performance bounds on greedy policies based on imperfect value functions. Technical Report NU-CCS-93-14, Northeastern University, College of Computer Science, Boston, MA, November.