Journal of Machine Learning Research 3 (2002) 59–72 Submitted 10/01; Published 7/02

On the Convergence of Optimistic Policy Iteration

John N. Tsitsiklis JNT@MIT.EDU
LIDS, Room 35-209
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139-4307, USA

Editor: Sridhar Mahadevan

Abstract

We consider a finite-state Markov decision problem and establish the convergence of a special case of optimistic policy iteration that involves Monte Carlo estimation of Q-values, in conjunction with greedy policy selection. We provide convergence results for a number of algorithmic variations, including one that involves temporal difference learning (bootstrapping) instead of Monte Carlo estimation. We also indicate some extensions that either fail or are unlikely to go through.

Keywords: Markov Decision Problem, Dynamic Programming, Reinforcement Learning, Monte Carlo, Stochastic Approximation, Temporal Differences.

1. Introduction

This paper deals with simulation-based methods for controlling stochastic systems or, in an alternative interpretation, learning methods for optimizing the policy of an agent interacting with its environment (Bertsekas & Tsitsiklis, 1996; Sutton & Barto, 1998). Many such methods aim at learning the optimal value function (the solution to the Bellman equation) associated with the problem of interest. Some rely on a model of the problem at hand, and some are model-free; some rely on a lookup table representation of the value function, and some employ parametric representations of the value function and value function approximation to combat the curse of dimensionality.

Developing an understanding of the convergence properties of such methods can be challenging. For this reason, one often starts by considering the more tractable case of lookup table representations (Watkins, 1989; Watkins & Dayan, 1992; Jaakkola, Jordan & Singh, 1994; Tsitsiklis, 1994). Even though the lookup table case is of limited practical importance, it sometimes serves as a stepping stone towards the understanding of the convergence properties of methods that incorporate value function approximation.

Within the realm of lookup table methods, Q-learning and TD(0) are based on "bootstrapping" and fall at one end of the spectrum of possible methods. At the other end of the spectrum, one has "Monte Carlo" methods that do not employ bootstrapping. Temporal difference methods, TD(λ), with $0 < \lambda < 1$ fall in the middle. Possibly the simplest Monte Carlo method is "Monte Carlo ES" (Sutton, 1999). In this method, one maintains and updates Q-values (one variable per state-action pair). The update is not based on the standard Q-learning formula. Instead, one observes the costs of complete trajectories, starting from a particular state-action pair, and lets the corresponding Q-value be the average of these costs. Sutton (1999) refers to the convergence of this method as an open problem.

In this paper, we deal exclusively with methods that make use of a lookup table representation. We settle the above mentioned open problem, for the case of a discounted cost criterion, under the assumption that every state-action pair is used to initialize the observed trajectories with the same frequency. A similar result is obtained for bootstrapping methods, based on TD(λ) for general λ. We remark that convergence is not guaranteed if the trajectory initializations are made in an arbitrary manner. We conclude with a brief discussion indicating that the convergence results are quite fragile with respect to the assumptions involved.

2. Preliminaries

We consider a Markov decision problem, with finite state and action sets. We are given a finite set $S = \{1, \ldots, n\}$ of states, and a finite set U of possible actions. With each state i and action u, we associate transition probabilities $p_{ij}(u)$ and one-stage costs $g(i, u)$. (We assume that one-stage costs are deterministic functions of i and u; the case of random rewards is discussed briefly in the last section.) We define a *policy* μ as a mapping $\mu : S \mapsto U$. Given any policy μ, the state evolution becomes a well-defined Markov chain X_t^μ with transition probabilities

$$P(X_{t+1}^\mu = j \mid X_t^\mu = i) = p_{ij}(\mu(i)).$$

The cost-to-go of policy μ starting from state i is defined as

$$J^\mu(i) = E\left[\sum_{t=0}^{\infty} \alpha^t g(X_t^\mu, \mu(X_t^\mu)) \mid X_0^\mu = i\right],$$

where α is a discount factor satisfying $0 < \alpha < 1$. The optimal cost-to-go function J^* is defined by

$$J^*(i) = \min_\mu J^\mu(i).$$

(The minimum is attained because the set of policies is finite.) The objective is to find an optimal policy, that is, a policy μ that attains the minimum in the above definition, simultaneously for all states i. Such an optimal policy is known to exist by standard results in dynamic programming.

We introduce some shorthand notation, along the lines of (Bertsekas & Tsitsiklis, 1996). Let P_μ be the $n \times n$ matrix of transition probabilities under policy μ. We will use the symbol J to indicate vectors of dimension n, to be interpreted as cost-to-go functions. In particular J^μ is the vector with components $J^\mu(i)$, and J^* is the vector with components $J^*(i)$. Let g_μ be the vector whose ith component is the one-stage cost $g(i, \mu(i))$ incurred by policy μ at state i. We finally introduce the dynamic programming operators $T, T_\mu : \Re^n \mapsto \Re^n$, which are defined as follows. For any vector $J \in \Re^n$, $T_\mu J$ is also a vector in $\Re^n$ whose ith component is given by

$$(T_\mu J)(i) = g(i, \mu(i)) + \alpha \sum_{j=1}^{n} p_{ij}(\mu(i)) J(j).$$

Similarly, the ith component of TJ is given by

$$(TJ)(i) = \min_u \left\{ g(i,u) + \alpha \sum_{j=1}^{n} p_{ij}(u) J(j) \right\}.$$

In vector notation, we have

$$T_\mu J = g_\mu + \alpha P_\mu J,$$

and

$$(TJ)(i) = \min_\mu (T_\mu J)(i), \qquad \forall\ i.$$

We use T^k to denote the composition of k copies of T. The notation T_μ^k is interpreted similarly.

As is well known, the vector J^μ is the unique solution to the linear equation

$$J^\mu = T_\mu J^\mu,$$

and also satisfies

$$J^\mu = \lim_{k\to\infty} T_\mu^k J, \qquad \forall\ J.$$

Also, the vector J^* is the unique solution to the Bellman equation

$$J^* = TJ^*.$$

A *greedy* policy corresponding to J is a policy μ in which, for each i, $\mu(i)$ is chosen to be a value of u that minimizes $g(i,u) + \alpha \sum_{j=1}^{n} p_{ij}(u)J(j)$. In vector notation, a greedy policy corresponding to J satisfies

$$T_\mu J = TJ.$$

If J is equal to J^*, then a greedy policy corresponding to J is guaranteed to be optimal.

We note some useful properties of the dynamic programming operators that will be used extensively. Let $e \in \Re^n$ be the vector with all components equal to 1. Then, for any scalar c, we have

$$T(J + ce) = TJ + \alpha ce, \qquad T_\mu(J+ce) = T_\mu J + \alpha ce.$$

Furthermore, the operators are monotone, in the sense that

$$J \le \overline{J} \implies TJ \le T\overline{J}, \;\; T_\mu J \le T_\mu \overline{J}.$$

Here and throughout the rest of the manuscript, a vector inequality such as $J \le \overline{J}$ is to be interpreted componentwise, i.e., $J(i) \le \overline{J}(i)$ for all i.

The *policy iteration* method operates as follows. Given a policy μ, one evaluates the vector J^μ (policy evaluation), and then chooses a new policy which is a greedy policy corresponding to J^μ (policy update). This iteration is repeated until the algorithm converges, which is guaranteed to happen after a finite number of steps, at which point an optimal policy is obtained. The computation of J^μ can take place by solving the system $J^\mu = T_\mu J^\mu$, if P_μ and g_μ are available, or it may involve a learning algorithm such as TD(λ). *Optimistic policy iteration* is a variation of ordinary policy iteration in which policy updates are carried out without waiting for policy evaluation to converge to J^μ. Such a method maintains a vector J, uses a greedy policy to generate a complete or partial trajectory, uses the results of the trajectory to carry out one iteration of an iterative policy evaluation algorithm (resulting in an update of J), and continues similarly. For example, if the partial trajectory consists of a single transition, and if policy evaluation involves a single value iteration update at the state before the transition, one obtains a form of asynchronous value iteration. The method considered in the next section is a particular type of optimistic policy iteration in which the policy evaluation method employed is Monte Carlo estimation.

3. Optimistic Policy Iteration Using Monte Carlo for Policy Evaluation

We start with a precise description of the algorithm to be analyzed. Let t be an iteration index. At each iteration t, we have available a vector J_t, and we let μ_t be a corresponding greedy policy, that is,

$$T_{\mu_t} J_t = T J_t.$$

For every state i, we simulate a trajectory that starts at state i and observe its cumulative discounted cost. (Note that this is only a "conceptual algorithm" because the trajectory will generally have to be of infinite length. Variations that correspond to implementable algorithms are discussed briefly in the last section.)

Since the expected cost of this trajectory is $J^{\mu_t}(i)$, the observed cumulative cost is equal to $J^{\mu_t}(i) + w_t(i)$, where $w_t(i)$ is zero-mean noise. We then update the vector J_t according to

$$J_{t+1}(i) = (1 - \gamma_t) J_t(i) + \gamma_t (J^{\mu_t}(i) + w_t(i)), \tag{1}$$

where γ_t is a (deterministic) scalar stepsize parameter.

In the special case where $J_0 = 0$ and $\gamma_t = 1/(t+1)$, it is easily verified that $J_t(i)$ is equal to the average of the observed cumulative costs of t independent trajectories that start at i. If the policy μ were held forever fixed, $J_t(i)$ would converge to $J^\mu(i)$. If a policy update were to be carried out only after an infinite number of updates of the form (1), the method would be identical to policy iteration. However, because the policy is continuously updated, we are dealing with an optimistic variant. Finally, we note that the method described here is *synchronous*, meaning that at each iteration we simultaneously observe n trajectories, one for each possible starting state.

The analysis of the method considered here is not entirely straightforward because it does not follow into the standard pattern of a contracting iteration perturbed by noise. Instead, one has to exploit the monotonicity properties of the dynamic programming operators T and T_μ.

Let $\mathcal{F}_t$ be the history of the algorithm up to and including the point where J_t has become available, but before simulating the trajectories that will determine the next update. Thus, w_t is a function of the random variables contained in $\mathcal{F}_{t+1}$, and

$$E[w_t(i) \mid \mathcal{F}_t] = 0.$$

Furthermore, the variance of $w_t(i)$ (conditioned on $\mathcal{F}_t$) is only a function of the current policy μ_t and the initial state. Since there are finitely many policies and states, the variance of $w_t(i)$ is bounded by some constant.

For the result that follows, as well as for all other results in this paper, we assume the usual stepsize conditions

$$\sum_{t=0}^{\infty} \gamma_t = \infty, \qquad \sum_{t=0}^{\infty} \gamma_t^2 < \infty.$$

Proposition 1 *The sequence J_t generated by the synchronous optimistic policy iteration algorithm (1), applied to a discounted problem, converges to J^*, with probability 1.*

Proof We define a scalar sequence c_t by letting

$$c_t = \max_i \left((TJ_t)(i) - J_t(i)\right).$$

The performance J^{μ_t} of a policy μ_t can be "worse" than the vector J_t that was used to produce that policy. The following lemma establishes a bound on the possible deterioration.

Lemma 2 *For every t, we have*

(a) $J^{\mu_t} \leq J_t + \dfrac{c_t}{1-\alpha} e,$

(b) $J^{\mu_t} \leq T_{\mu_t} J_t + \dfrac{\alpha c_t}{1-\alpha} e,$

(c) $T_{\mu_t}^k J_t \leq J_t + \dfrac{c_t}{1-\alpha} e,$ *for all k,*

where e is the vector with all components equal to 1.

Proof From the definition of μ_t, we have $T_{\mu_t} J_t = TJ_t$. Using also the definition of c_t, we have $T_{\mu_t} J_t \leq J_t + c_t e$. We apply T_{μ_t} to both sides of the latter inequality, to obtain

$$T_{\mu_t}^2 J_t \leq T_{\mu_t}(J_t + c_t e) = T_{\mu_t} J_t + \alpha c_t e \leq J_t + c_t e + \alpha c_t e.$$

Continuing inductively, we obtain

$$T_{\mu_t}^k J_t \leq J_t + (1 + \alpha + \cdots + \alpha^{k-1}) c_t e \leq J_t + \frac{c_t}{1-\alpha} e, \qquad \forall\, k,$$

which proves part (c). Since $T_{\mu_t}^k J_t$ converges to J^{μ_t}, part (a) follows.

To prove part (b), we apply T_{μ_t} to both sides of the result of part (a) and use the fact $T_{\mu_t} J^{\mu_t} = J^{\mu_t}$. ∎

If we had $c_t \leq 0$, we would obtain $J^{\mu_t} \leq J_t$, and in the absence of noise, $J_{t+1} \leq J_t$. In the presence of such monotonicity, convergence is easy to establish. In the remainder of the proof, we will use standard tools for analyzing stochastic iterations to show that the effects of the noise are asymptotically negligible, and also that c_t converges to zero or less, which brings us to the easier case mentioned above.

Recall that

$$T_{\mu_t} J = g_{\mu_t} + \alpha P_{\mu_t} J, \qquad \forall\ J.$$

Using this affine property of T_{μ_t}, we have

$$\begin{aligned}
TJ_{t+1} &\leq T_{\mu_t} J_{t+1} \\
&= T_{\mu_t}((1-\gamma_t)J_t + \gamma_t J^{\mu_t} + \gamma_t w_t) \\
&= g_{\mu_t} + (1-\gamma_t)\alpha P_{\mu_t} J_t + \gamma_t \alpha P_{\mu_t} J^{\mu_t} + \gamma_t \alpha P_{\mu_t} w_t \\
&= (1-\gamma_t) T_{\mu_t} J_t + \gamma_t T_{\mu_t} J^{\mu_t} + \gamma_t \alpha P_{\mu_t} w_t \\
&= (1-\gamma_t) T J_t + \gamma_t J^{\mu_t} + \gamma_t w_t + \gamma_t \alpha P_{\mu_t} w_t - \gamma_t w_t \\
&= (1-\gamma_t) J_t + (1-\gamma_t)(TJ_t - J_t) + \gamma_t J^{\mu_t} + \gamma_t w_t + \gamma_t \alpha P_{\mu_t} w_t - \gamma_t w_t \\
&= J_{t+1} + (1-\gamma_t)(TJ_t - J_t) + \gamma_t v_t,
\end{aligned} \tag{2}$$

where $v_t = \alpha P_{\mu_t} w_t - w_t$. Note that $E[v_t \mid \mathcal{F}_t] = 0$ and

$$E[\|v_t\|^2 \mid \mathcal{F}_t] \leq CE[\|w_t\|^2 \mid \mathcal{F}_t] \leq CA,$$

for some constants A and C.

We have established so far that

$$TJ_{t+1} - J_{t+1} \leq (1-\gamma_t)(TJ_t - J_t) + \gamma_t v_t.$$

Let us define $X_t = TJ_t - J_t$ and note that

$$X_{t+1} \leq (1-\gamma_t) X_t + \gamma_t v_t.$$

We will compare X_t to the sequence of vectors Y_t defined by $Y_0 = X_0$ and

$$Y_{t+1} = (1-\gamma_t) Y_t + \gamma_t v_t.$$

An easy inductive argument shows that $X_t \leq Y_t$ for all t. Using standard results on convergence of stochastic iterations, e.g., Example 4.3 in p. 143 of (Bertsekas & Tsitsiklis, 1996), Y_t converges to zero, with probability 1. (The qualifier "with probability 1" will be omitted in the sequel but should be understood to be apply whenever limits of random variables are involved.) Consequently,

$$\limsup_{t\to\infty} X_t \leq 0.$$

Since $c_t = \max_i X_t(i)$, we conclude that for every $\epsilon > 0$, there exists a time $t(\epsilon)$ such that

$$c_t \leq \epsilon, \qquad \forall\ t \geq t(\epsilon).$$

Using Lemma 2(b) and the fact $TJ_t = T_{\mu_t} J_t$, we obtain

$$J^{\mu_t} \leq TJ_t + \frac{\alpha c_t}{1-\alpha} e \leq TJ_t + \frac{\epsilon\alpha}{1-\alpha} e, \qquad \forall\ t \geq t(\epsilon).$$

Thus, for $t \geq t(\epsilon)$, we have

$$\begin{aligned} J_{t+1} &= (1-\gamma_t)J_t + \gamma_t J^{\mu_t} + \gamma_t w_t \\ &\leq (1-\gamma_t)J_t + \gamma_t T J_t + \gamma_t \frac{\epsilon\alpha}{1-\alpha} e + \gamma_t w_t. \end{aligned}$$

Let us fix $\epsilon > 0$. We will carry out a comparison of the sequence J_t with the sequence Z_t defined by $Z_{t(\epsilon)} = J_{t(\epsilon)}$ and

$$Z_{t+1} = (1-\gamma_t)Z_t + \gamma_t T Z_t + \gamma_t \frac{\epsilon\alpha}{1-\alpha} e + \gamma_t w_t, \qquad \forall\ t \geq t(\epsilon). \tag{3}$$

An easy inductive argument shows that $J_t \leq Z_t$ for all $t \geq t(\epsilon)$.

We define a mapping $H_\delta : \Re^n \mapsto \Re^n$ by letting

$$H_\delta Z = TZ + \delta e.$$

Given that T is a contraction mapping with respect to the maximum norm, it follows that the mapping H_δ is also a contraction mapping. Furthermore, the unique fixed point Z_δ^* of H_δ is equal to $J^* + \delta e/(1-\alpha)$, because

$$H_\delta\left(J^* + \frac{\delta e}{1-\alpha}\right) = T\left(J^* + \frac{\delta e}{1-\alpha}\right) + \delta e = J^* + \frac{\alpha\delta e}{1-\alpha} + \delta e = J^* + \frac{\delta e}{1-\alpha}.$$

Note that the iteration (3) is of the form

$$Z_{t+1} = (1-\gamma_t)Z_t + \gamma_t(H_\delta Z_t + w_t),$$

with $\delta = \alpha\epsilon/(1-\alpha)$. Given that H_δ is a maximum-norm contraction and the noise w_t is zero mean and with bounded variance, Prop. 4.4 in p. 156 of (Bertsekas & Tsitsiklis, 1996) shows that Z_t must converge to Z_δ^*. Recall that $J_t \leq Z_t$ for all $t \geq t(\epsilon)$. It follows that $\limsup_{t\to\infty} J_t \leq Z_\delta^*$. Since ϵ can be chosen arbitrarily close to 0, the same is true for δ, and we conclude that

$$\limsup_{t\to\infty} J_t \leq \inf_{\delta>0} Z_\delta^* = J^*.$$

Finally, we use the relation $J^{\mu_t} \geq J^*$ to obtain

$$J_{t+1} \geq (1-\gamma_t)J_t + \gamma_t J^* + \gamma_t w_t.$$

We have $J_t \geq V_t$, where V_t satisfies $V_0 = J_0$ and

$$V_{t+1} = (1-\gamma_t)V_t + \gamma_t J^* + \gamma_t w_t.$$

The sequence V_t converges to J^*, we obtain $\liminf_{t\to\infty} J_t \geq J^*$, and the proof is complete. ∎

We now discuss another variant for which convergence is similarly established. At each iteration t, instead of generating a trajectory from *every* initial state, let us pick a *single* state i, randomly, uniformly, and independently from everything else, and generate a single trajectory starting from i. We then update the cost-to-go function only at state i. If n is the cardinality of the state space, we have at each iteration,

$$J_{t+1}(i) = \begin{cases} (1-\gamma_t)J_t(i) + \gamma_t J^{\mu_t}(i) + \gamma_t w_t(i), & \text{with probability } 1/n, \\ J_t(i), & \text{otherwise.} \end{cases}$$

It is not hard to see that this algorithm can be described in the form

$$J_{t+1}(i) = \Big(1 - \frac{\gamma_t}{n}\Big) J_t(i) + \frac{\gamma_t}{n}\Big(J^{\mu_t}(i) + w_t(i) + v_t(i)\Big), \tag{4}$$

where $v_t(i)$ is a noise term, reflecting the randomness in the choice of i. In particular,

$$v_t(i) = (n\chi_t(i) - 1)\big(- J_t(i) + J^{\mu_t}(i) + w_t(i)\big),$$

where the $\chi_t(i)$ are random variables such that $\chi_t(i) = 1$ if state i is selected, and $\chi_t(i) = 0$, otherwise. Because a state is selected uniformly, the expected value of $\chi_t(i)$ is $1/n$, which implies that $E[v_t(i) \mid \mathcal{F}_t] = 0$. Furthermore, because there are finitely many possible policies and states, $J^{\mu_t}(i)$ is bounded, which implies that

$$E[\|v_t\|^2 \mid \mathcal{F}_t] \le A + B\|J_t\|^2,$$

for some constants A and B. Using these observations, the proof of Prop. 1 goes through with small modifications. The only difference is that the conditional variance of v_t is not bounded by a constant, but by a quadratic in $\|J_t\|$. Even so, we are still within the setting considered in Sections 4.2-4.3 of Bertsekas & Tsitsiklis (1996), and the various stochastic approximation results quoted in the course of the proof of Prop. 1 remain applicable. (See the proof of Prop. 3 in the next section, for the specifics of the technique used to handle the absence of a bound on the conditional variance.)

An extension that does not seem possible, at least with this particular proof method, concerns the case where the initial state of a trajectory is picked at random, as just discussed, but according to a nonuniform distribution. Effectively, this replaces the scalar stepsize γ_t/n in Eq. (4) by a component-dependent stepsize $\gamma_t p(i)$, where $p(i)$ is the probability that state i is selected. Tracing back the proof of Prop. 1, the scalar stepsize γ_t has to be replaced by a diagonal matrix Γ_t. We note that the equalities in Eq. (2) essentially rely on the property $T_\mu((1-\gamma)J + \gamma\overline{J}) = (1-\gamma)T_\mu J + \gamma T_\mu \overline{J}$. However, this property fails to hold when γ is replaced by a diagonal matrix Γ, because Γ and P_μ need not commute.

On the other hand, the following slight variation does lead to a convergent algorithm. Allow the algorithm to select the initial state according to nonuniform probabilities $p(i)$ but instead of having a deterministic stepsize γ_t, use a component-dependent stepsize of the form $\gamma_t(i) = 1/n_t(i)$, where $n_t(i)$ is the number of trajectories, out of the first t trajectories, for which i was selected to be the initial state. In the long run, $n_t(i)$ will be equal to $p(i)t$, plus $O(\sqrt{t})$ terms, so that $\gamma_t(i) \sim (1/p(i)t)$. Multiplying by the probability $p(i)$ that state i is selected and that $J(i)$ is updated, we see that the expected change in $J(i)$ is proportional to $p(i) \cdot 1/(p(i)t) = 1/t$. This is the same mathematical structure as in the proof of Prop. 1, and the result remains valid.

In a last and interesting variant, a single initial state is chosen at random, but an update of $J(i)$ is carried out for every state i that is visited by the trajectory. This is possible, because the cost accumulated by the tail end of the trajectory, starting from the time that i is visited, provides an unbiased estimate of $J^\mu(i)$. There are different implementations of this variation, depending on whether multiple visits of a trajectory to the same state i lead to multiple updates ("every visit" version) or to a single one ("first visit" version) (Singh & Sutton, 1996). With either implementation, the probability that $J(i)$ is updated during an iteration is nonuniform and convergence is an open question, for the same reasons as those in our earlier discussion of the nonuniform case.

Even though we are not able to settle the convergence problem for the case of nonuniform choice of initial states, we do know that some form of statistical regularity is needed. If one allows the selection of the initial state to be arbitrary (subject only to the condition that every state is selected infinitely often), Example 5.12 in pp. 234-236 of Bertsekas & Tsitsiklis (1996) provides a counterexample to convergence.

4. Optimistic Synchronous Policy Iteration based on TD(λ)

In this section, we extend the result of the preceding section, to cover a related method that uses TD(λ) for policy evaluation, instead of Monte Carlo.

In temporal difference methods, one simulates a trajectory $i_0, i_1, \ldots$, starting from an initial state $i = i_0$, using a policy μ, and records the temporal differences

$$d_k = g(i_k, \mu(i_k)) + \alpha J(i_{k+1}) - J(i_k).$$

Each temporal difference contributes an update increment to $J(i)$. The total update is of the form

$$J(i) := J(i) + \gamma \sum_{k=0}^{\infty} \alpha^k \lambda^k d_k, \tag{5}$$

where $\lambda \in [0, 1)$. An equivalent form of this update rule, obtained after some algebra, is

$$J(i) := (1-\gamma)J(i) + \gamma(1-\lambda) \sum_{k=0}^{\infty} \lambda^k \Big(g(i_0) + \alpha g(i_1) + \cdots + \alpha^k g(i_k) + \alpha^{k+1} J(i_{k+1}) \Big).$$

In vector notation, we have

$$J := (1-\gamma)J + \gamma(1-\lambda) \sum_{k=0}^{\infty} \lambda^k T_\mu^{k+1} J + \gamma w,$$

where w is a zero-mean noise vector reflecting the difference between the observed temporal differences and their expected values. Note that if we set $\lambda = 1$ in Eq. (5), the update rule becomes

$$J(i) := (1-\gamma)J(i) + \gamma \sum_{k=0}^{\infty} \alpha^k g(i_k),$$

and we recover the method of the preceding section.

If we let $\lambda = 0$, the update rule (5) becomes

$$J := (1-\gamma)J + \gamma T_\mu J + w.$$

If furthermore, μ is chosen to be a greedy policy corresponding to J, we obtain the update rule

$$J := (1-\gamma)J + \gamma TJ + w.$$

Since T is a maximum-norm contraction, general results apply (Prop. 4.4 in p. 156 of Bertsekas & Tsitsiklis, 1996), and show that the method will converge to J^*, even if carried out asynchronously (that is, even if initial states of trajectories are chosen in an unstructured manner, as long as each state is selected infinitely often). When λ is close to zero, one expects that the same convergence result will still go through, by a "continuity" argument. For general values of λ, however, this proof technique does not seem adequate for establishing asynchronous convergence. We will therefore restrict once more to a synchronous version.

Similar to the preceding section, at each iteration t, we have available a vector J_t, and we let μ_t be a corresponding greedy policy, that is,

$$T_{\mu_t} J_t = TJ_t.$$

For every state i, we simulate a trajectory that starts at state i, calculate temporal differences, and carry out the update prescribed by (5), which translates to

$$J_{t+1} = (1-\gamma_t)J_t + \gamma_t(1-\lambda)\sum_{k=0}^{\infty}\lambda^k T_{\mu_t}^{k+1} J_t + \gamma_t w_t, \tag{6}$$

where γ_t is a (deterministic) scalar stepsize parameter.

For some insight into the mathematical issues that arise with this algorithm, think of the term $T_{\mu_t}^{k+1} J_t$ as being of the form $T_{\mu(J)}^{k+1} J$, where $\mu(J)$ is a greedy policy associated with J. For $k=0$, we have $T_{\mu(J)}J = TJ$, and the mapping $J \mapsto TJ$ is a contraction (hence our earlier argument for the case $\lambda = 0$). However, for positive k, the mapping $J \mapsto T_{\mu(J)}^{k+1} J$ is far from being a contraction, and is in fact discontinuous: small changes in J can result in a different policy $\mu(J)$ and hence in large changes in the value of $T_{\mu(J)}^{k+1} J$. Thus, arguments based on contraction properties are not applicable.

Proposition 3 *The sequence J_t generated by the synchronous optimistic TD(λ) algorithm (6), applied to a discounted problem, converges to J^*, with probability 1.*

Proof Note that $E[w_t \mid \mathcal{F}_t] = 0$. Furthermore, since the update term in Eq. (5) depends linearly on J, it is seen that

$$E[\|w_t\|^2 \mid \mathcal{F}_t] \le A + B\|J_t\|^2,$$

for some constants A and B.

Lemma 4 *The sequence J_t is bounded, with probability 1.*

Proof The update equation (6) is of the form

$$J_{t+1} = (1-\gamma_t)J_t + \gamma_t H_t J_t + \gamma_t w_t,$$

where the mapping H_t has the property

$$\|H_t J_t\|_\infty \le \max_{k\ge 0} \|T_{\mu_t}^{k+1} J_t\|_\infty \le \alpha \|J_t\|_\infty + D,$$

for some constant D. (Here, we use the maximum norm $\|\cdot\|_\infty$ defined by $\|J\|_\infty = \max_i |J(i)|$.) The boundedness of the sequence J_t follows from Prop. 4.7 in p. 159 of Bertsekas & Tsitsiklis (1996). ∎

We now continue in a manner that parallels the proof of Prop. 1, and using again the notation $c_t = \max_i ((TJ_t)(i) - J_t(i))$. The chain of equalities in Eq. (2) is replaced by the following calculation:

$$\begin{aligned}
TJ_{t+1} &\le T_{\mu_t} J_{t+1} \\
&= T_{\mu_t}\Big((1-\gamma_t)J_t + \gamma_t(1-\lambda)\sum_{k=0}^{\infty} \lambda^k T_{\mu_t}^{k+1} J_t + \gamma_t w_t\Big) \\
&= g_{\mu_t} + (1-\gamma_t)\alpha P_{\mu_t} J_t + \gamma_t \alpha P_{\mu_t}(1-\lambda)\sum_{k=0}^{\infty} \lambda^k T_{\mu_t}^{k+1} J_t + \gamma_t \alpha P_{\mu_t} w_t \\
&= (1-\gamma_t) T_{\mu_t} J_t + \gamma_t(1-\lambda) T_{\mu_t} \sum_{k=0}^{\infty} \lambda^k T_{\mu_t}^{k+1} J_t + \gamma_t \alpha P_{\mu_t} w_t \\
&\le (1-\gamma_t)J_t + (1-\gamma_t)(TJ_t - J_t) + \gamma_t(1-\lambda)\sum_{k=0}^{\infty} \lambda^k T_{\mu_t}^{k+1}(J_t + c_t e) + \gamma_t \alpha P_{\mu_t} w_t \\
&= (1-\gamma_t)J_t + (1-\gamma_t)(TJ_t - J_t) + \gamma_t(1-\lambda)\sum_{k=0}^{\infty} \lambda^k (T_{\mu_t}^{k+1} J_t + \alpha^{k+1} c_t e) + \gamma_t \alpha P_{\mu_t} w_t \\
&= J_{t+1} + (1-\gamma_t)(TJ_t - J_t) + \gamma_t(1-\lambda)\sum_{k=0}^{\infty} \lambda^k \alpha^{k+1} c_t e + \gamma_t \alpha P_{\mu_t} w_t - \gamma_t w_t \\
&\le J_{t+1} + (1-\gamma_t)(TJ_t - J_t) + \gamma_t \alpha c_t e + \gamma_t v_t,
\end{aligned}$$

where $v_t = \alpha P_{\mu_t} w_t - w_t$.

We have established so far that

$$TJ_{t+1} - J_{t+1} \le (1-\gamma_t)(TJ_t - J_t) + \gamma_t \alpha c_t e + \gamma_t v_t.$$

Let us define $X_t = TJ_t - J_t$ and note that

$$X_{t+1} \le (1-\gamma_t)X_t + \gamma_t \alpha e \max_i X_t(i) + \gamma_t v_t.$$

We will compare X_t to the sequence of vectors Y_t defined by $Y_0 = X_0$ and

$$Y_{t+1} = (1-\gamma_t)Y_t + \gamma_t \alpha e \max_i Y_t(i) + \gamma_t v_t.$$

An easy inductive argument shows that $X_t \le Y_t$ for all t. We notice that the mapping $Y \mapsto \alpha e \max_i Y(i)$ is a maximum norm contraction. Fix a positive integer l, and consider the stopped process $v^l(t)$ which coincides with v_t as long as $E[v_t^2(i) \mid \mathcal{F}_t] \le l$, and is equal to 0 thereafter. Consider the iteration

$$Y_{t+1}^l = (1-\gamma_t)Y_t^l + \gamma_t \alpha e \max_i Y_t^l(i) + \gamma_t v_t^l.$$

Using results on convergence of stochastic iterations involving contraction mappings (see, e.g., Prop. 4.4 in p. 156 of Bertsekas & Tsitsiklis, 1996), Y_t^l converges to zero, for every l. By Lemma 4, the sequence J_t is bounded, which implies that the the sequence $E[v_t^2(i) \mid \mathcal{F}_t]$ is also bounded. Therefore, with probability 1, there exists some l such that $v_t^l = v_t$ for all t, and, consequently, $Y_t^l = Y_t$ for all t. Hence, Y_t also converges to zero, which implies that

$$\limsup_{t\to\infty} X_t \le 0.$$

As in the proof of Prop. 1, we fix some $\epsilon > 0$, and choose $t(\epsilon)$ such that

$$c_t \le \epsilon, \qquad \forall\ t \ge t(\epsilon).$$

By Lemma 2(c), we have $T_{\mu_t}^k J_t \le T_{\mu_t} J_t + \epsilon e/(1-\alpha)$, and Eq. (6) yields

$$J_{t+1} \le (1-\gamma_t)J_t + \gamma_t T J_t + \gamma_t \frac{\epsilon\alpha}{1-\alpha} e + \gamma_t w_t, \qquad t \ge t(\epsilon).$$

From here on, the rest of the proof is identical to the last part of the proof of Prop. 1. ■

5. The Model-Free Case

The algorithms of the preceding two sections require knowledge of the system model. This is because g_μ and P_μ are needed in order to generate a greedy policy μ_t corresponding to the current vector J_t. But of course, if a model is available, learning methods with lookup table representations are uninteresting, except to the extent that they provide insights into more general settings.

However, even in the absence of a model, a related method based on Q-values is applicable. The method is as follows. (We only describe it for the Monte Carlo case. The reader should have no difficulty extending this discussion to the case of general λ.) For every state-action pair (i,u), we introduce a Q-value $Q(i,u)$ which is an estimate of the cost-to-go starting from state i, given that the first decision has been fixed to be u.

At each iteration t, we have available a vector Q_t, with components $Q_t(i,u)$, and we let μ_t be a corresponding greedy policy, that is, for every i we select $\mu_t(i)$ to be a value of u that results in the smallest $Q_t(i,u)$. For every pair (i,u), we generate a trajectory that starts at state i, chooses u as the first decision, and follows the policy μ_t thereafter. Let $Q^{\mu_t}(i,u)$ be the cumulative expected cost of such a trajectory. The observed cost is of the form $Q^{\mu_t}(i,u) + w_t(i,u)$, where $w_t(i,u)$ is a zero-mean noise term. We then update Q according to

$$Q_{t+1}(i,u) = (1-\gamma_t)Q_t(i,u) + \gamma_t(Q^{\mu_t}(i,u) + w_t(i,u)), \qquad \forall\ (i,u). \tag{7}$$

where γ_t is a deterministic scalar stepsize parameter. The specific question raised by Sutton (1999) is whether Q_t converges to Q^*, where $Q^*(i,u) = \min_\mu Q^\mu(i,u)$, because when $Q = Q^*$, a corresponding greedy policy is known to be optimal.

Proposition 5 *The sequence Q_t, generated by the algorithm (7), applied to a discounted problem, converges to Q^*, with probability 1.*

Proof This is the special case of the result in Prop. 1, applied to a new problem in which the state space consists of all "regular" states i in the original problem, together with all state-action pairs (i, u). The dynamics in the new problem are as follows: when at a regular state i, one selects a decision u and moves deterministically to the state (i, u). When at a state of the form (i, u), there are no decisions to be made, the cost $g(i, u)$ is incurred, and the next state is j with probability $p_{ij}(u)$. A cost-to-go vector for this new problem has two kinds of components: those of the form $J(i)$ for regular states i, and those of the form $Q(i, u)$ for the new states (i, u).

Let us now apply the algorithm of Section 3 to this new problem. At time t, we have available vectors J_t and Q_t. A greedy policy μ_t is determined, which prescribes the actions to be taken at regular states i by considering the values of $Q_t(i, u)$ for various u. Trajectories are simulated under μ_t starting from every regular state i and from every new state (i, u). The results $Q^{\mu_t}(i, u) + w_t(i, u)$ (where $w_t(i, u)$ is the usual simulation noise term) of the trajectories starting at new states are used to update Q according to Eq. (7), which takes the form

$$Q_{t+1} = (1 - \gamma_t)Q_t + \gamma_t Q^{\mu_t} + \gamma_t w_t.$$

(The vector J_t is also updated, but this has no effect on the Q_t or on the greedy policies.) By Prop. 1, Q_t converges to Q^*. We then recognize that the algorithm we have just analyzed is mathematically identical (as far as the Q-values are concerned) to the one described by Eq. (7). ∎

6. Discussion and Conclusions

As a practical matter, the algorithms considered in earlier sections are not implementable because each iteration requires the generation of infinitely long trajectories. This difficulty can be bypassed in a few different ways. One possibility is to only generate trajectories over a long enough horizon, say of duration T, where α^T is very close to zero, so that the cost accumulated during the finite trajectory is a very close approximation to the infinite trajectory cost. Another possibility is to restrict to problems that have a zero-cost absorbing state that is guaranteed to be reached eventually. In that case, we only need to generate trajectories for a finite (though random) time, that is, until the absorbing state is reached. A last possibility, which is always applicable, is to let the process terminate with probability α at each stage, and to accumulate undiscounted costs. This is justified because the expected undiscounted cost until termination is equal to the expected infinite horizon discounted cost.

The formulation that we have used assumes that the cost per stage $g(i, u)$ is a deterministic function of i and u. In a slightly more general model, one can assume that the one-stage cost is a random variable whose conditional expectation (given the past history of the process) is equal to $g(i, u)$, and whose conditional variance is bounded. Our results have straightforward extensions to this case, because the "noise" in the one-stage costs can be incorporated into the zero-mean noise term w_t in either of the update equations we have considered (Eqs. 1, 6, or 7).

We have provided a number of results and have settled one of the open problems by Sutton (1999), on the convergence of Monte Carlo based optimistic policy iteration. However, these results seem to be quite fragile. For example, unlike Q-learning with lookup table representations, the methods considered here are known to be nonconvergent in the presence of asynchronism. It is still an open question whether convergence can be established if the various states are selected with some regularity (e.g., at random, but according to a fixed – nonuniform – distribution), or if one considers the "every-visit" version (Singh & Sutton) of the algorithm. Another open question (probably not as hard), is to extend the results to undiscounted (stochastic shortest path) problems, under the assumption that termination is inevitable. Finally, there is the more interesting question of what happens in the presence of function approximation. Here, we do not see much ground for optimism.

Acknowledgments

The author is grateful to Rich Sutton for reviving his interest in this problem. This research was partially supported by the NSF under contract ECS-9873451 and by the AFOSR under contract F49620-99-1-0320.

References

[1] Bertsekas, D. P., & Tsitsiklis, J. N. *Neuro-Dynamic Programming.* Athena Scientific, Belmont, Massachusetts, 1996.

[2] Jaakkola, T., Jordan, M. I., & Singh, S. P. On the Convergence of Stochastic Iterative Dynamic Programming Algorithms. *Neural Computation*, 6:1185-1201, 1994.

[3] Sutton, R. S. Open Theoretical Questions in Reinforcement Learning. In Fischer, P., & Simon, H.U. (Eds.), *Proceedings of the Fourth European Conference on Computational Learning Theory (Proceedings EuroCOLT'99)*, pages 11-17, 1999. Springer-Verlag. `ftp://ftp.cs.umass.edu/pub/anw/pub/sutton/sutton-99.ps`

[4] Sutton, R. S., & Barto, A. G. *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, Massachusetts, 1998.

[5] Singh, S. P., & Sutton, R. S. Reinforcement Learning with Replacing Eligibility Traces, *Machine Learning*, 22:123-158, 1996.

[6] Tsitsiklis, J. N. Asynchronous Stochastic Approximation and Q-Learning. *Machine Learning,* 16:185-202, 1994.

[7] Watkins, C. J. C. H., & Dayan, P. Q-Learning. *Machine Learning,* 8:279-292, 1992.

[8] Watkins, C. J. C. H. Learning from Delayed Rewards. Ph.D. thesis, Cambridge University, Cambridge, England, 1989.