Neural Computation, Vol. 6, No. 6, 1994

On the Convergence of Stochastic Iterative Dynamic Programming Algorithms*

Tommi Jaakkola
Michael I. Jordan
Department of Brain and Cognitive Sciences
Massachusetts Institute of Technology

Satinder P. Singh
Department of Computer Science
University of Massachusetts at Amherst

Abstract

Recent developments in the area of reinforcement learning have yielded a number of new algorithms for the prediction and control of Markovian environments. These algorithms, including the TD(λ) algorithm of Sutton (1988) and the Q-learning algorithm of Watkins (1989), can be motivated heuristically as approximations to dynamic programming (DP). In this paper we provide a rigorous proof of convergence of these DP-based learning algorithms by relating them to the powerful techniques of stochastic approximation theory via a new convergence theorem. The theorem establishes a general class of convergent algorithms to which both TD(λ) and Q-learning belong.

*This project was supported in part by a grant from the McDonnell-Pew Foundation, by a grant from ATR Human Information Processing Research Laboratories, by a grant from Siemens Corporation, and by grant N00014-90-J-1942 from the Office of Naval Research. The project was also supported by NSF grant ASC-9217041 in support of the Center for Biological and Computational Learning at MIT, including funds provided by DARPA under the HPCC program. Michael I. Jordan is a NSF Presidential Young Investigator.

An important component of many real world learning problems is the *temporal credit assignment problem*—the problem of assigning credit or blame to individual components of a temporally-extended plan of action, based on the success or failure of the plan as a whole. To solve such a problem, the learner must be equipped with the ability to assess the long-term consequences of particular choices of action and must be willing to forego an immediate payoff for the prospect of a longer term gain. Moreover, because most real world problems involving prediction of the future consequences of actions involve substantial uncertainty, the learner must be prepared to make use of a probability calculus for assessing and comparing actions.

There has been increasing interest in the temporal credit assignment problem, due principally to the development of learning algorithms based on the theory of dynamic programming (DP) (Barto, Sutton, & Watkins, 1990; Werbos, 1992). Sutton's (1988) TD(λ) algorithm addressed the problem of learning to predict in a Markov environment, utilizing a temporal difference operator to update the predictions. Watkins' (1989) Q-learning algorithm extended Sutton's work to control problems, and also clarified the ties to dynamic programming.

In the current paper, our concern is with the stochastic convergence of DP-based learning algorithms. Although Watkins (1989) and Watkins and Dayan (1992) proved that Q-learning converges with probability one, and Dayan (1992) observed that TD(0) is a special case of Q-learning and therefore also converges with probability one, these proofs rely on a construction that is particular to Q-learning and fail to reveal the ties of Q-learning to the broad theory of stochastic approximation (e.g., Wasan, 1969). Our goal here is to provide a simpler proof of convergence for Q-learning by making direct use of stochastic approximation theory. We also show that our proof extends to TD(λ) for arbitrary λ. Several other authors have recently presented results that are similar to those presented here: Dayan and Sejnowski (1993) for TD(λ), Peng and Williams (1993) for TD(λ), and Tsitsiklis (1993) for Q-learning. Our results appear to be closest to those of Tsitsiklis (1993).

We begin with a general overview of Markov decision problems and DP. We introduce the Q-learning algorithm as a stochastic form of DP. We then present a proof of convergence for a general class of stochastic processes of which Q-learning is a special case. We then discuss TD(λ) and show that it is also a special case of our theorem.

Markov decision problems

A useful mathematical model of temporal credit assignment problems, studied in stochastic control theory (Aoki, 1967) and operations research (Ross, 1970), is the *Markov decision problem*. Markov decision problems are built on the formalism of controlled Markov chains. Let $S = 1, 2, \ldots, N$ be a discrete state space and let $U(i)$ be the discrete set of actions available to the learner when the chain is in state i. The probability of making a transition from state i to state j is given by $p_{ij}(u)$, where

$u \in U(i)$. The learner defines a *policy* μ, which is a function from states to actions. Associated with every policy μ is a Markov chain defined by the state transition probabilities $p_{ij}(\mu(i))$.

There is an *instantaneous cost* $c_i(u)$ associated with each state i and action u, where $c_i(u)$ is a random variable with expected value $\bar{c}_i(u)$. We also define a *value function* $V_\mu(i)$, which is the expected sum of discounted future costs given that the system begins in state i and follows policy μ:

$$V_\mu(i) = \lim_{N \to \infty} E\{\sum_{t=0}^{N-1} \gamma^t c_{s_t}(\mu(s_t)) | s_0 = i\}, \tag{1}$$

where $s_t \in S$ is the state of the Markov chain at time t. Future costs are discounted by a factor γ^t, where $\gamma \in (0,1)$. We wish to find a policy that minimizes the value function:

$$V^*(i) = \min_\mu V_\mu(i). \tag{2}$$

Such a policy is referred to as an *optimal policy* and the corresponding value function is referred to as the *optimal value function*. Note that the optimal value function is unique, but an optimal policy need not be unique. (For example, if the costs are independent of the actions and states all policies are optimal.)

Markov decision problems can be solved by dynamic programming (Bertsekas, 1987). The basis of the DP approach is an equation that characterizes the optimal value function. This equation, known as Bellman's equation, characterizes the optimal value of the state in terms of the optimal values of possible successor states:

$$V^*(i) = \min_{u \in U(i)} \{\bar{c}_i(u) + \gamma \sum_{j \in S} p_{ij}(u) V^*(j)\}. \tag{3}$$

To motivate Bellman's equation, suppose that the system is in state i at time t and consider how $V^*(i)$ should be characterized in terms of possible transitions out of state i. Suppose that action u is selected and the system transitions to state j. The expression $c_i(u) + \gamma V^*(j)$ is the cost of making a transition out of state i plus the discounted cost of following an optimal policy thereafter. The minimum of the expected value of this expression, over possible choices of actions, seems a plausible measure of the optimal cost at i and by Bellman's equation is indeed equal to $V^*(i)$.

There are a variety of computational techniques available for solving Bellman's equation. The technique that we focus on in the current paper is a iterative algorithm known as *value iteration*. Value iteration solves for $V^*(i)$ by setting up a recurrence relation for which Bellman's equation is a fixed point. Denoting the estimate of $V^*(i)$ at the k^{th} iteration as $V^{(k)}(i)$, we have:

$$V^{(k+1)}(i) = \min_{u \in U(i)} \{\bar{c}_i(u) + \gamma \sum_{j \in S} p_{ij}(u) V^{(k)}(j)\} \tag{4}$$

This iteration can be shown to converge to $V^*(i)$ for arbitrary initial $V^{(0)}(i)$ (Bertsekas, 1987). The proof is based on showing that the iteration from $V^{(k)}(i)$ to $V^{(k+1)}(i)$ is a contraction mapping. That is, it can be shown that:

$$\max_i |V^{(k+1)}(i) - V^*(i)| \leq \gamma \max_i |V^{(k)}(i) - V^*(i)|, \tag{5}$$

which implies that $V^{(k)}(i)$ converges to $V^*(i)$ and also places an upper bound on the convergence rate.

Watkins (1989) utilized an alternative notation for expressing Bellman's equation that is particularly convenient for deriving learning algorithms. Define the function $Q^*(i,u)$ to be the expression appearing inside the "min" operator of Bellman's equation:

$$Q^*(i,u) = \bar{c}_i(u) + \gamma \sum_{j \in S} p_{ij}(u) V^*(j) \tag{6}$$

Using this notation Bellman's equation can be written as follows:

$$V^*(i) = \min_{u \in U(i)} Q^*(i,u). \tag{7}$$

Moreover, value iteration can be expressed in terms of Q functions:

$$Q^{(k+1)}(i,u) = \bar{c}_i(u) + \gamma \sum_{j \in S} p_{ij}(u) V^{(k)}(j), \tag{8}$$

where $V^{(k)}(i)$ is defined in terms of $Q^{(k)}(i,u)$ as follows:

$$V^{(k)}(i) = \min_{u \in U(i)} Q^{(k)}(i,u). \tag{9}$$

The mathematical convenience obtained from using Q's rather than V's derives from the fact that the minimization operator appears inside the expectation in Equation 8, whereas it appears outside the the expectation in Equation 4. This fact plays an important role in the convergence proof presented in this paper.

The value iteration algorithm in Equation 4 or Equation 8 can also be executed *asynchronously* (Bertsekas & Tsitsiklis, 1989). In an asynchronous implementation, the update of the value of a particular state may proceed independently of the updates of the values of other states. Bertsekas & Tsitsiklis (1989) show that as long as each state is updated infinitely often and each action is tried an infinite number of times in each state, then the asynchronous algorithm eventually converges to the optimal value function. Moreover, asynchronous execution has the advantage that it is directly applicable to *real-time* Markov decision problems (RTDP; Barto, Bradtke, & Singh, 1993). In a real-time setting, the system uses its evolving value function to choose control actions for an actual process and updates the values of the states along the trajectory followed by the process.

Dynamic programming serves as a starting point for deriving a variety of learning algorithms for systems that interact with Markov environments (Barto, Bradtke, & Singh, 1993; Sutton, 1988; Watkins, 1989). Indeed, real-time dynamic programming is arguably a form of learning algorithm as it stands. Although RTDP requires that the system possess a complete model of the environment (i.e., the probabilities $p_{ij}(u)$ and the expected costs $\bar{c}_i(u)$ are assumed known), the performance of a system using RTDP improves over time, and its improvement is focused on the states that are actually visited. The system "learns" by transforming knowledge in one format (the model) into another format (the value function).

A more difficult learning problem arises when the probabilistic structure of the environment is unknown. There are two approaches to dealing with this situation (cf. Barto, Bradtke, & Singh, 1993). An *indirect* approach acquires a model of the environment incrementally, by estimating the costs and the transition probabilities, and then uses this model in an ongoing DP computation. A *direct* method dispenses with constructing a model and attempts to estimate the optimal value function (or the optimal Q-values) directly. In the remainder of this paper, we focus on direct methods, in particular the Q-learning algorithm of Watkins (1989) and the TD(λ) algorithm of Sutton (1988).

The Q-learning algorithm is a stochastic form of value iteration. Consider Equation 8, which expresses the update of the Q values in terms of the Q values of successor states. To perform a step of value iteration requires knowing the expected costs and the transition probabilities. Although such a step cannot be performed without a model, it is nonetheless possible to *estimate* the appropriate update. For an arbitrary V function, the quantity $\sum_{j \in S} p_{ij}(u)V(j)$ can be estimated by the quantity $V(j)$, if successor state j is chosen with probability $p_{ij}(u)$. But this is assured by simply following the transitions of the actual Markov environment, which makes a transition from state i to state j with probability $p_{ij}(u)$. Thus the sample value of V at the successor state is an unbiased estimate of the sum. Moreover $c_i(u)$ is an unbiased estimate of $\bar{c}_i(u)$. This reasoning leads to the following relaxation algorithm, where we use $Q_t(i, u)$ and $V_t(i)$ to denote the learner's estimates of the Q function and V function at time t, respectively:

$$Q_{t+1}(s_t, u_t) = (1 - \alpha_t(s_t, u_t))Q_t(s_t, u_t) + \alpha_t(s_t, u_t)[c_{s_t}(u_t) + \gamma V_t(s_{t+1})] \tag{10}$$

where

$$V_t(s_{t+1}) = \min_{u \in U(s_{t+1})} Q_t(s_t, u_t). \tag{11}$$

The variables $\alpha_t(s_t, u_t)$ are zero except for the state that is being updated at time t.

The fact that Q-learning is a stochastic form of value iteration immediately suggests the use of stochastic approximation theory, in particular the classical framework of Robbins and Monro (1951). Robbins-Monro theory treats the stochastic convergence of a sequence of unbiased estimates of a regression function, providing conditions under which the sequence converges to a root of the function. Although the stochastic convergence of Q-learning is not an immediate consequence of Robbins-Monro theory, the theory does provide results that can be adapted to studying the convergence of DP-based learning algorithms. In this paper we utilize a result from Dvoretzky's (1956) formulation of Robbins-Monro theory to prove the convergence of both Q-learning and TD(λ).

Convergence proof for Q-learning

Our proof is based on the observation that the Q-learning algorithm can be viewed as a stochastic process to which techniques of stochastic approximation are generally applicable. Due to the lack of a formulation of stochastic approximation for the maximum norm, however, we need to slightly extend the standard results. This is accomplished by the following theorem the proof of which is given in Appendix A.

Theorem 1 *A random iterative process* $\Delta_{n+1}(x) = (1 - \alpha_n(x))\Delta_n(x) + \beta_n(x)F_n(x)$ *converges to zero w.p.1 under the following assumptions:*

1) $x \in S$, *where* S *is a finite set.*

2) $\sum_n \alpha_n(x) = \infty$, $\sum_n \alpha_n^2(x) < \infty$, $\sum_n \beta_n(x) = \infty$, $\sum_n \beta_n^2(x) < \infty$, *and* $\mathrm{E}\{\beta_n(x)|P_n\} \leq \mathrm{E}\{\alpha_n(x)|P_n\}$ *uniformly over* x *w.p.1.*

3) $\| \mathrm{E}\{F_n(x)|P_n, \beta_n\} \|_W \leq \gamma \| \Delta_n \|_W$, *where* $\gamma \in (0, 1)$.

4) $\mathrm{Var}\{F_n(x)|P_n, \beta_n\} \leq C(1 + \| \Delta_n \|_W)^2$, *where* C *is some constant.*

Here $P_n = \{X_n, X_{n-1}, \ldots, F_{n-1}, \ldots, \alpha_{n-1}, \ldots, \beta_{n-1}, \ldots\}$ *stands for the past at step* n. $F_n(x)$, $\alpha_n(x)$ *and* $\beta_n(x)$ *are allowed to depend on the past.* $\alpha_n(x)$ *and* $\beta_n(x)$ *are assumed to be non-negative and mutually independent given* P_n. *The notation* $\| \cdot \|_W$ *refers to some weighted maximum norm.*

In applying the theorem, the Δ_n process will generally represent the difference between a stochastic process of interest and some optimal value (e.g., the optimal value function). The formulation of the theorem therefore requires knowledge to be available about the optimal solution to the learning problem before it can be applied to any algorithm whose convergence is to be verified. In the case of Q-learning the required knowledge is available through the theory of DP and Bellman's equation in particular.

The convergence of the Q-learning algorithm now follows easily by relating the algorithm to the converging stochastic process defined by Theorem 1.[1] In the form of the theorem we have:

[1]We note that the theorem is more powerful than is needed to prove the convergence of Q-learning. Its generality, however, allows it to be applied to other algorithms as well (see the following section on TD(λ).

Theorem 2 *The Q-learning algorithm given by*

$$Q_{t+1}(s_t,u_t) = (1-\alpha_t(s_t,u_t))Q_t(s_t,u_t) + \alpha_t(s_t,u_t)[c_{s_t}(u_t) + \gamma V_t(s_{t+1})]$$

converges to the optimal $Q^*(s,u)$ *values if*

1) *The state and action spaces are finite.*

2) $\sum_t \alpha_t(s,u) = \infty$ *and* $\sum_t \alpha_t^2(s,u) < \infty$ *uniformly over* s *and* u *w.p.1.*

3) $\mathrm{Var}\{c_s(u)\}$ *is finite.*

3) *If* $\gamma = 1$ *all policies lead to a cost free terminal state w.p.1.*

Proof. By subtracting $Q^*(s,u)$ from both sides of the learning rule and by defining $\Delta_t(s,u) = Q_t(s,u) - Q^*(s,u)$ together with

$$F_t(s,u) = c_s(u) + \gamma V_t(s_{next}) - Q^*(s,u) \tag{12}$$

the Q-learning algorithm can be seen to have the form of the process in Theorem 1 with $\beta_t(s,u) = \alpha_t(s,u)$.

To verify that $F_t(s,u)$ has the required properties we begin by showing that it is a contraction mapping with respect to some maximum norm. This is done by relating F_t to the DP value iteration operator for the same Markov chain. More specifically,

$$\begin{aligned}
\max_u |\mathrm{E}\{F_t(i,u)\}| &= \gamma \max_u |\sum_j p_{ij}(u)[V_t(j) - V^*(j)]| \\
&\leq \gamma \max_u \sum_j p_{ij}(u) \max_v |Q_t(j,v) - Q^*(j,v)| \\
&= \gamma \max_u \sum_j p_{ij}(u) V^{\Delta}(j) = T(V^{\Delta})(i)
\end{aligned}$$

where we have used the notation $V^{\Delta}(j) = \max_v |Q_t(j,v) - Q^*(j,v)|$ and T is the DP value iteration operator for the case where the costs associated with each state are zero (cf. Equation (4)). If $\gamma < 1$ the contraction property of $\mathrm{E}\{F_t(i,u)\}$ can be seen from the fourth formula by bounding $\sum_j p_{ij}(u)V^{\Delta}(j)$ by $\max_j V^{\Delta}(j)$ and then including the γ factor. When the future costs are not discounted ($\gamma = 1$) but the chain is absorbing and all policies lead to the terminal state w.p.1 there still exists a weighted maximum norm with respect to which T is a contraction mapping (see e.g. Bertsekas & Tsitsiklis, 1989) thereby forcing the contraction of $\mathrm{E}\{F_t(i,u)\}$. The variance of $F_t(s,u)$ given the past is within the bounds of Theorem 1 as it depends on $Q_t(s,u)$ at most linearly and the variance of $c_s(u)$ is bounded.

Note that the proof covers both the on-line and batch versions. □

The TD(λ) algorithm

The TD(λ) (Sutton, 1988) is also a DP-based learning algorithm that is naturally defined in a Markov environment. Unlike Q-learning, however, TD does not involve decision-making tasks but rather predictions about the future costs of an evolving system. TD(λ) converges to the same predictions as a version of Q-learning in which there is only one action available at each state, but the algorithms are derived from slightly different grounds and their behavioral differences are not well understood. In this section we introduce the algorithm and its derivation. The proof of convergence is given in the following section.

Let us define $V_t(i)$ to be the current estimate of the expected cost incurred during the evolution of the system starting from state i and let c_i denote the instantaneous random cost at state i. As in the case of Q-learning we assume that the future costs are discounted at each state by a factor γ. If no discounting takes place ($\gamma = 1$) we need to assume that the Markov chain is absorbing, that is, there exists a cost free terminal state to which the system converges with probability one.

We are concerned with estimating the future costs that the learner has to incur. One way to achieve these predictions is to simply observe n consecutive random costs weighted by the discount factor and to add the best estimate of the costs thereafter. This gives us the estimate

$$V_t^{(n)}(i_t) = c_{i_t} + \gamma c_{i_{t+1}} + \gamma^2 c_{i_{t+2}} + \ldots + \gamma^{n-1} c_{i_{t+n-1}} + \gamma^n V_t(i_{t+n}) \tag{13}$$

The expected value of this can be shown to be a strictly better estimate than the current estimate is (Watkins, 1989). In the undiscounted case this holds only when n is larger than some chain-dependent constant. To demonstrate this let us replace V_t with V^* in the above formula giving $\mathrm{E}\{V_t^{*(n)}(i_t)\} = V^*(i_t)$ which implies

$$\max_i |\mathrm{E}\{V_t^{(n)}(i)\} - V^*(i)| \leq \gamma^n \max_i \Pr\{m_i \geq n\} \max_i |V_t(i) - V^*(i)| \tag{14}$$

where m_i is the number of steps in a sequence that begins in state i (infinite in the non-absorbing case). This implies that if either $\gamma < 1$ or n is large enough so that the chain can terminate before n steps starting from an arbitrary initial state then the estimate $V_t^{(n)}$ is strictly better than V_t. In general, the larger n the more unbiased the estimate is as the effect of incorrect V_t vanishes. However, larger n increases the variance of the estimate as there are more (independent) terms in the sum.

Despite the error reduction property of the truncated estimate it is difficult to calculate in practice as one would have to wait n steps before the predictions could be updated. In addition it clearly has a huge variance. A remedy to these problems is obtained by constructing a new estimate by averaging over the truncated predictions. TD(λ) is based on taking the geometric average:

$$V_t^{\lambda}(i) = (1-\lambda) \sum_{n=1}^{\infty} \lambda^{n-1} V_t^{(n)}(i) \tag{15}$$

As a weighted average it is still a strictly better estimate than $V_t(i)$ with the additional benefit of being better in the undiscounted case as well (as the summation extends to infinity). Furthermore, we have introduced a new parameter λ which affects the trade-off between the bias and variance of the estimate (Watkins, 1989). An increase in λ puts more weight on less biased estimates with higher variances and thus the bias in V_t^λ decreases at the expense of a higher variance.

The mathematical convenience of using the geometric average can be seen as follows. Given the estimates $V_t^\lambda(i)$ the obvious way to use them in a learning rule is

$$V_{t+1}(i_t) = V_t(i_t) + \alpha[V_t^\lambda(i_t) - V_t(i_t)] \tag{16}$$

In terms of *prediction differences*, that is

$$\Delta_t(i_t) = c_{i_t} + \gamma V_t(i_{t+1}) - V_t(i_t) \tag{17}$$

the geometric weighting allows us to write the correction term in the learning rule as

$$V_t^\lambda(i_t) - V_t(i_t) = \Delta_t(i_t) + (\lambda\gamma)\Delta_t(i_{t+1}) + (\lambda\gamma)^2\Delta_t(i_{t+2}) + \ldots \tag{18}$$

Note that up to now the prediction differences that need to be calculated in the future depend on the current $V_t(i)$. If the chain is nonabsorbing this computational implausibility can, however, be overcome by updating the predictions at each step with the prediction differences calculated by using the current predictions. This procedure gives the on-line version of TD(λ):

$$V_{t+1}(i) = V_t(i) + \alpha_t \Delta_t(i_t) \sum_{k=0}^{t} (\gamma\lambda)^{t-k} \chi_i(k) \tag{19}$$

where $\chi_i(k)$ is the indicator variable of whether state i was visited at k^{th} step (of a sequence). Note that the sum contains the effect of the modifications or activity traces initiated at past time steps. Moreover, it is important to note that in this case the theoretically desirable properties of the estimates derived earlier may hold only asymptotically (see the convergence proof in the next section).

In the absorbing case the estimates $V_t(i)$ can also be updated off-line, that is, after a complete sequence has been observed. The learning rule for this case is derived simply from collecting the correction traces initiated at each step of the sequence. More concisely, the total correction is the sum of individual correction traces illustrated in Equation (18). This results in the batch learning rule

$$V_{n+1}(i) = V_n(i) + \alpha_n \sum_{t=1}^{m} \Delta_n(i_t) \sum_{k=0}^{t} (\gamma\lambda)^{t-k} \chi_i(k) \tag{20}$$

where the $(m+1)^{th}$ step is the termination state.

We note that the above derivation of the TD(λ) algorithm corresponds to the specific choice of a linear representation for the predictors $V_t(i)$ (see, e.g., Dayan, 1992). Learning rules for other representations can be obtained using gradient descent but these are not considered here. In practice TD(λ) is usually applied to an absorbing chain thus allowing the use of either the batch or the on-line version but the latter is usually preferred.

Convergence of TD(λ)

As we are interested in strong forms of convergence we need to modify the algorithm slightly. The learning rate parameters α_n are replaced by $\alpha_n(i)$ which satisfy $\sum_n \alpha_n(i) = \infty$ and $\sum_n \alpha_n^2(i) < \infty$ uniformly w.p.1. These parameters allow asynchronous updating and they can, in general, be random variables. The convergence of the algorithm is guaranteed by the following theorem which is an application of Theorem 1.

Theorem 3 *For any finite absorbing Markov chain, for any distribution of starting states with no inaccessible states, and for any distributions of the costs with finite variances the TD(λ) algorithm given by*

1)

$$V_{n+1}(i) = V_n(i) + \alpha_n(i) \sum_{t=1}^{m} [c_{i_t} + \gamma V_n(i_{t+1}) - V_n(i_t)] \sum_{k=1}^{t} (\gamma\lambda)^{t-k} \chi_i(k)$$

$\sum_n \alpha_n(i) = \infty$ *and* $\sum_n \alpha_n^2(i) < \infty$ *uniformly over* i *w.p.1.*

2)

$$V_{t+1}(i) = V_t(i) + \alpha_t(i)[c_{i_t} + \gamma V_t(i_{t+1}) - V_t(i_t)] \sum_{k=1}^{t} (\gamma\lambda)^{t-k} \chi_i(k)$$

$\sum_t \alpha_t(i) = \infty$ *and* $\sum_n \alpha_t^2(i) < \infty$ *uniformly w.p.1 and within sequences* $\alpha_t(i)/max_{t\in S}\alpha_t(i) \rightarrow 1$ *uniformly over* i *w.p.1.*

converges to the optimal predictions w.p.1 provided $\gamma, \lambda \in [0,1]$ *with* $\gamma\lambda < 1$.

Proof for (1): Using the ideas described in the previous section the learning rule can be written as

$$\begin{aligned} V_{n+1}(i) &= V_n(i) + \alpha_n(i)[G_n(i) - \frac{m(i)}{\mathrm{E}\{m(i)\}} V_n(i)] \\ G_n(i) &= \frac{1}{\mathrm{E}\{m(i)\}} \sum_{k=1}^{m(i)} V_n^{\lambda}(i;k) \end{aligned}$$

where $V_n^\lambda(i;k)$ is an estimate calculated at the k^{th} occurence of state i in a sequence and for mathematical convenience we have made the transformation $\alpha_n(i) \rightarrow \mathrm{E}\{m(i)\}\alpha_n(i)$, where $m(i)$ is the number of times state i was visited during the sequence.

To apply Theorem 1 we subtract $V^*(i)$, the optimal predictions, from both sides of the learning equation. By identifying $\alpha_n(i) := \alpha_n(i)m(i)/\mathrm{E}\{m(i)\}$, $\beta_n(i) := \alpha_n(i)$, and $F_n(i) := G_n(i) - V^*(i)m(i)/\mathrm{E}\{m(i)\}$ we need to show that these satisfy the conditions of Theorem 1. For $\alpha_n(i)$ and $\beta_n(i)$ this is obvious. We begin here by showing that $F_n(i)$ indeed is a contraction mapping. To this end,

$$
\begin{aligned}
&\max_i |\mathrm{E}\{F_n(i) \mid V_n\}| = \\
&\quad \max_i |\frac{1}{\mathrm{E}\{m(i)\}} \mathrm{E}\{(V_n^\lambda(i;1) - V^*(i)) + (V_n^\lambda(i;2) - V^*(i)) + \ldots \mid V_n\}|
\end{aligned}
$$

which can be bounded above by using the relation

$$
\begin{aligned}
&|\mathrm{E}\{V_n^\lambda(i;k) - V^*(i) \mid V_n\}| \\
&\quad \leq \mathrm{E}\Big\{ |\mathrm{E}\{V_n^\lambda(i;k) - V^*(i) \mid m(i) \geq k, V_n\}|\theta(m(i)-k) \mid V_n\Big\} \\
&\quad \leq P\{m(i) \geq k\}|\mathrm{E}\{V_n^\lambda(i) - V^*(i) \mid V_n\}| \\
&\quad \leq \gamma P\{m(i) \geq k\} \max_i |V_n(i) - V^*(i)|
\end{aligned}
$$

where $\theta(x) = 0$ if $x < 0$ and 1 otherwise. Here we have also used the fact that $V_n^\lambda(i)$ is a contraction mapping independent of possible discounting. As $\sum_k P\{m(i) \geq k\} = \mathrm{E}\{m(i)\}$ we finally get

$$
\max_i |\mathrm{E}\{F_n(i) \mid V_n\}| \leq \gamma \max_i |V_n(i) - V^*(i)|
$$

The variance of $F_n(i)$ can be seen to be bounded by

$$
\mathrm{E}\{m^4\} \max_i |V_n(i)|^2
$$

For any absorbing Markov chain the convergence to the terminal state is geometric and thus for every finite k, $\mathrm{E}\{m^k\} \leq C(k)$, implying that the variance of $F_n(i)$ is within the bounds of Theorem 1. As Theorem 1 is now applicable we can conclude that the batch version of TD(λ) converges to the optimal predictions w.p.1. $\square$

Proof for (2) The proof for the on-line version is achieved by showing that the effect of the on-line updating vanishes in the limit thereby forcing the two versions to be equal asymptotically. We view the on-line version as a batch algorithm in which the updates are made after each complete sequence but are made in such a manner so as to be equal to those made on-line.

Define $G_n^{'}(i) = G_n(i) + G_n^{\Delta}(i)$ to be a new batch estimate taking into account the on-line updating within sequences. Here $G_n(i)$ is the batch estimate with the desired properties (see the proof for (1)) and $G_n^{\Delta}(i)$ is the difference between the two. We take the new batch learning parameters to be the maxima over a sequence, that is $a_n(i) = \max_{t\in S} \alpha_t(i)$. As all the $\alpha_t(i)$ satisfy the required conditions uniformly w.p.1 these new learning parameters satisfy them as well.

To characterize the new batch estimate we consider the on-line updating iteration and decompose it into three parallel processes. Let $V_t^B(i)$ be the value function due to the real batch updating with $a_n(i)$ as learning parameters, let $V_t^{\Delta}(i)$ be the difference between this batch value function and the value function resulting from the on-line updating and let $R_t(i)$ be the change in the value function after t on-line updating steps. These processes can be written as

$$\begin{aligned}
V_{t+1}^B(i) &= V_t^B(i) + a_n(i)[c_{i_t} + \gamma V_n(i_{t+1}) - V_n(i_t)]\beta_t(i) \\
V_{t+1}^{\Delta}(i) &= V_t^{\Delta}(i) + \alpha_t(i)[c_{i_t} + \gamma V_n(i_{t+1}) - V_n(i_t) + \gamma R_t(i_{t+1}) - R_t(i_t)]\beta_t(i) \\
&\quad - a_n(i)[c_{i_t} + \gamma V_n(i_{t+1}) - V_n(i_t)]\beta_t(i) \\
R_{t+1}(i) &= R_t(i) + \alpha_t(i)[c_{i_t} + \gamma V_n(i_{t+1}) - V_n(i_t) + \gamma R_t(i_{t+1}) - R_t(i_t)]\beta_t(i)
\end{aligned}$$

with initial conditions $V_0^B(i) = V_n(i)$, $V_0^{\Delta}(i) = R_0(i) = 0$. By denoting $C_t^* = |c_{i_t} + \gamma V^*(i_{t+1}) - V^*(i_t)|$ and $\Delta_n = \| V_n - V^* \|$ and bounding the learning parameters by their norms and by the largest member of the sequence $(a_n(i))$ we readily obtain

$$\begin{aligned}
\mathrm{E}\{|V_{t+1}^{\Delta}(i)|\} &\leq \mathrm{E}\{|V_t^{\Delta}(i)|\} + \frac{2}{1-\gamma\lambda} a_n(i) \left\{ \| \frac{\alpha_t}{a_n} - 1 \| (\Delta_n + \mathrm{E}\{C_t^*\}) + \mathrm{E}\{\| R_t \|\} \right\} \\
\mathrm{E}\{\| R_{t+1} \|\} &\leq \mathrm{E}\{\| R_t \|\} + \frac{2}{1-\gamma\lambda} \| a_n \| (\Delta_n + \mathrm{E}\{C_t^*\} + \mathrm{E}\{\| R_t \|\}).
\end{aligned}$$

Let us define $C = 2/(1-\gamma\lambda)$ and $C^* \geq \mathrm{E}\{C_t^*\}$. Note that such a C^* can always be found as the optimal value function is bounded under the assumptions of the theorem. By iterating we get

$$\mathrm{E}\{\| R_m \|\} \leq C(\Delta_n + C^*)[(1 + C \| a_n \|)^m - 1]$$

and inserting this into the $\mathrm{E}\{|V_t^{\Delta}(i)|\}$ iteration and noting that it is only a sum we get an upper bound

$$\mathrm{E}\{|V_m^{\Delta}(i)|\} \leq a_n(i) m C^2 (\Delta_n + C^*) \left\{ \max_t \| \frac{\alpha_t}{a_n} - 1 \| + (1 + C \| a_n \|)^m - 1 \right\} \quad (21)$$

As the new batch updates are made using $a_n(i)$ parameters the quantity of interest to us is $\mathrm{E}\{\mathrm{E}\{|V_m^{\Delta}(i)|\}/a_n(i)\}_m$ which is a bound for $\mathrm{E}\{G_n^{\Delta}(i)\}$. In order to use Theorem 1 we need to show that the estimates $G_n^{'}$ have the desired properties. In other words, as the $G_n(i)$ have these properties we need to show that the effect of $G_n^{\Delta}(i)$ is vanishing.

Let us first consider the contraction property of the estimate. In any absorbing Markov chain the probability of absorption increases geometrically and thus the probability of having m steps in a sequence can be bounded by $C_p \gamma_p^m$ where $\gamma_p < 1$. Due to the convergence of $a_n(i)$, for n sufficiently large ($a_n(i)$ small enough), the expectation of $\mathrm{E}\{\| R_m \|\}$ over m is bounded and goes to zero in the limit w.p.1 (all the terms contain a_n^k where $k \geq 1$). As also $\| \frac{\alpha_t}{a_n} - 1 \|$ converges to zero w.p.1, $\mathrm{E}\{G_n^{\Delta}(i)\}$ vanishes in the limit (see Equation (21)). Using Equation (21) we can write these results as

$$\begin{aligned} \| \mathrm{E}\{G_n^{'} - V^*\} \| &\leq \| \mathrm{E}\{G_n - V^*\} \| + \| G_n^{\Delta} \| \\ &\leq (\gamma^{'} + C_n^1) \| V_n - V^* \| + C_n^2 \end{aligned}$$

where C_n^1 and C_n^2 go to zero with w.p.1. This implies that for any $\epsilon > 0$ and $\| V_n - V^* \| \gg \epsilon$ there exists $\gamma < 1$ such that

$$\| \mathrm{E}\{G_n^{'} - V^*\} \| \leq \gamma \| V_n - V^* \|$$

for n large enough.

The variance of the estimate satisfies condition (4) of Theorem 1. To see this, note that in deriving the bound for $\mathrm{E}\{G_n^{\Delta}(i)\}$ the only random variables were C_t^*, the variances of which are bounded, and m, the variance of which can be bounded exactly in the same way as the expectation; it too vanishes in the limit.

Theorem 1 now guarantees that for any ϵ the value function in the on-line algorithm converges w.p.1 into some ϵ-bounded region of V^* and therefore the algorithm itself converges to V^* w.p.1. □

Conclusions

In this paper we have extended results from stochastic approximation theory to cover asynchronous relaxation processes which have a contraction property with respect to some maximum norm (Theorem 1). This new class of converging iterative processes is shown to include both the Q-learning and TD(λ) algorithms in either their on-line or batch versions. We note that the convergence of the on-line version of TD(λ) has not been shown previously. We also wish to emphasize the simplicity of our results. The convergence proofs for Q-learning and TD(λ) utilize only high-level statistical properties of the estimates used in these algorithms and do not rely on constructions specific to the algorithms. Our approach also sheds additional light on the similarities between Q-learning and TD(λ).

Although Theorem 1 is readily applicable to DP-based learning schemes, the theory of Dynamic Programming is important only for its characterization of the optimal solution and for a contraction property needed in applying the theorem. The theorem can be applied to iterative algorithms of different types as well.

Finally we note that Theorem 1 can be extended to cover processes that do not show the usual contraction property thereby increasing its applicability to algorithms of possibly more practical importance.

Proof of Theorem 1

In this section we provide a detailed proof of the theorem on which the convergence proofs for Q-learning and TD(λ) were based. We introduce and prove three essential lemmas, which will also help to clarify ties to the literature and the ideas behind the theorem, followed by the proof of Theorem 1. The notation $\| \cdot \|_W = \max_x | \cdot / W(x)|$ will be used in what follows.

Lemma 1 *A random process*

$$w_{n+1}(x) = (1 - \alpha_n(x))w_n(x) + \beta_n(x)r_n(x).$$

converges to zero with probability one if the following conditions are satisfied:

1) $\sum_n \alpha_n(x) = \infty$, $\sum_n \alpha_n^2(x) < \infty$, $\sum_n \beta_n(x) = \infty$, *and* $\sum_n \beta_n^2(x) < \infty$ *uniformly over* x *w.p.1.*

2) $\mathrm{E}\{r_n(x)|P_n, \beta_n\} = 0$ *and* $\mathrm{E}\{r_n^2(x)|P_n, \beta_n\} \leq C$ *w.p.1, where*

$$P_n = \{w_n, w_{n-1}, \ldots, r_{n-1}, r_{n-2}, \ldots, \alpha_{n-1}, \alpha_{n-2}, \ldots, \beta_{n-1}, \beta_{n-2}, \ldots\}$$

All the random variables are allowed to depend on the past P_n. $\alpha_n(x)$ *and* $\beta_n(x)$ *are assumed to be non-negative and mutually independent given* P_n.

Proof. Except for the appearance of $\beta_n(x)$ this is a standard result. With the above definitions convergence follows directly from Dvoretzky's extended theorem (Dvoretzky, 1956).

Lemma 2 *Consider a stochastic iteration*

$$X_{n+1}(x) = G_n(X_n, Y_n, x)$$

where G_n *is a sequence of functions and* Y_n *is a random process. Let* $(\Omega, \mathcal{F}, P)$ *be a probability space and assume that the process is scale invariant, that is, w.p.1 for all* $\omega \in \Omega$

$$G(\beta X_n, Y_n(\omega), x) = \beta G(X_n, Y_n(\omega), x)$$

Assume further that if we kept $\| X_n \|$ *bounded by scaling, then* X_n *would converge to zero w.p.1. These assumptions are sufficient to guarantee that the original process converges to zero w.p.1.*

Proof. Note that multiplying X_n by β corresponds to having initialized the process with βX_0. Now fix some constant C. If during the iteration, $\| X_n \|$ increases above C, then X_n is scaled so that $\| X_n \| = C$. By the second assumption then this process must converge w.p.1. To show that the net effect of the corrections must stay finite w.p.1 we note that if $\| X_n \|$ converges then for any $\epsilon > 0$ there exists M_ϵ such that $\| X_n \| < \epsilon < C$ for all $n > M_\epsilon$ with probability at least $1 - \epsilon$. But this implies that the iteration stays below C after M_ϵ and converges to zero without any further corrections. $\square$

Lemma 3 *A stochastic process $X_{n+1}(x) = (1 - \alpha(x))X_n(x) + \gamma\beta_n(x) \| X_n \|$ converges to zero w.p.1 provided*

1) *$x \in S$, where S is a finite set.*

2) *$\sum_n \alpha_n(x) = \infty$, $\sum_n \alpha_n^2(x) < \infty$, $\sum_n \beta_n(x) = \infty$, $\sum_n \beta_n^2(x) < \infty$, and $\mathrm{E}\{\beta_n(x)|P_n\} \leq \mathrm{E}\{\alpha_n(x)|P_n\}$ uniformly over x w.p.1.*

where

$$P_n = \{X_n, X_{n-1}, \ldots, \alpha_{n-1}, \alpha_{n-2}, \ldots, \beta_{n-1}, \beta_{n-2}, \ldots\}$$

$\alpha_n(x)$ and $\beta_n(x)$ are assumed to be non-negative and mutually independent given P_n.

Proof. Essentially the proof is an application of Lemma 2. To this end, assume that we keep $\| X_n \| \leq C_1$ by scaling which allows the iterative process to be bounded by

$$|X_{n+1}(x)| \leq (1 - \alpha_n(x))|X_n(x)| + \gamma\beta_n(x)C_1$$

This is linear in $|X_n(x)|$ and can be easily shown to converge w.p.1 to some $X^*(x)$, where $\| X^* \| \leq \gamma C_1$. Hence, for small enough ϵ, there exists $M_1(\epsilon)$ such that $\| X_n \| \leq C_1/(1+\epsilon)$ for all $n > M_1(\epsilon)$ with probability at least $p_1(\epsilon)$. With probability $p_1(\epsilon)$ the procedure can be repeated for $C_2 = C_1/(1+\epsilon)$. Continuing in this manner and choosing $p_k(\epsilon)$ so that $\prod_k p_k(\epsilon)$ goes to one as $\epsilon \to 0$ we obtain the w.p.1 convergence of the bounded iteration and Lemma 2 can be applied. $\square$

Theorem 1 *A random iterative process $\Delta_{n+1}(x) = (1 - \alpha_n(x))\Delta_n(x) + \beta_n(x)F_n(x)$ converges to zero w.p.1 under the following assumptions:*

1) *$x \in S$, where S is a finite set.*

2) *$\sum_n \alpha_n(x) = \infty$, $\sum_n \alpha_n^2(x) < \infty$, $\sum_n \beta_n(x) = \infty$, $\sum_n \beta_n^2(x) < \infty$, and $\mathrm{E}\{\beta_n(x)|P_n\} \leq \mathrm{E}\{\alpha_n(x)|P_n\}$uniformly over x w.p.1.*

3) *$\| \mathrm{E}\{F_n(x)|P_n, \beta_n\} \|_W \leq \gamma \| \Delta_n \|_W$, where $\gamma \in (0, 1)$.*

4) *$\mathrm{Var}\{F_n(x)|P_n, \beta_n\} \leq C(1 + \| \Delta_n \|_W)^2$, where C is some constant.*

Here $P_n = \{X_n, X_{n-1}, \ldots, F_{n-1}, \ldots, \alpha_{n-1}, \ldots, \beta_{n-1}, \ldots\}$ stands for the past at step n. $F_n(x)$, $\alpha_n(x)$ and $\beta_n(x)$ are allowed to depend on the past. $\alpha_n(x)$ and $\beta_n(x)$ are assumed to be non-negative and mutually independent given P_n. The notation $\| \cdot \|_W$ refers to some weighted maximum norm.

Proof. By defining $r_n(x) = F_n(x) - \mathrm{E}\{F_n(x)|P_n, \beta_n\}$ we can decompose the iterative process into two parallel processes given by

$$\begin{aligned} \delta_{n+1}(x) &= (1-\alpha_n(x))\delta_n(x) + \beta_n(x)\mathrm{E}\{F_n(x)|P_n, \beta_n\} \\ w_{n+1}(x) &= (1-\alpha_n(x))w_n(x) + \beta_n(x)r_n(x) \end{aligned} \tag{22}$$

where $\Delta_n(x) = \delta_n(x) + w_n(x)$. Dividing the equations by $W(x)$ for each x and denoting $\delta_n^{'}(x) = \delta_n(x)/W(x)$, $w_n^{'}(x) = w_n(x)/W(x)$, and $r_n^{'}(x) = r_n(x)/W(x)$ we can bound the $\delta_n^{'}$ process by assumption 3) and rewrite the equation pair as

$$\begin{aligned} |\delta_{n+1}^{'}(x)| &\leq (1-\alpha_n(x))|\delta_n^{'}(x)| + \gamma\beta_n(x) \parallel |\delta^{'}| + w_n^{'} \parallel \\ w_{n+1}^{'}(x) &= (1-\alpha_n(x))w_n^{'}(x) + \gamma\beta_n(x)r_n^{'}(x) \end{aligned}$$

Assume for a moment that the Δ_n process stays bounded. Then the variance of $r_n^{'}(x)$ is bounded by some constant C and thereby $w_n^{'}$ converges to zero w.p.1 according to Lemma 1. Hence, there exists M such that for all $n > M \parallel w_n^{'} \parallel < \epsilon$ with probability at least $1-\epsilon$. This implies that the $\delta_n^{'}$ process can be further bounded by

$$|\delta_{n+1}^{'}(x)| \leq (1-\alpha_n(x))|\delta_n^{'}(x)| + \gamma\beta_n(x) \parallel \delta_n^{'} + \epsilon \parallel$$

with probability $> 1-\epsilon$. If we choose C such that $\gamma(C+1)/C < 1$ then for $\parallel \delta_n^{'} \parallel > C\epsilon$

$$\gamma \parallel \delta_n^{'} + \epsilon \parallel \leq \gamma(C+1)/C \parallel \delta_n^{'} \parallel$$

and the process defined by this upper bound converges to zero w.p.1 by Lemma 3. Thus $\parallel \delta_n^{'} \parallel$ converges w.p.1 to some value bounded by $C\epsilon$ which guarantees the w.p.1 convergence of the original process under the boundedness assumption.

By assumption (4) $r_n^{'}(x)$ can be written as $(1+ \parallel \delta_n + w_n \parallel)s_n(x)$, where $\mathrm{E}\{s_n^2(x)|P_n\} \leq C$. Let us now decompose w_n as $u_n + v_n$ with

$$u_{n+1}(x) = (1-\alpha_n(x))u_n(x) + \gamma\beta_n(x) \parallel \delta_n^{'} + u_n + v_n \parallel s_n(x)$$

and v_n converges to zero w.p.1 by Lemma 1. Again by choosing C such that $\gamma(C+1)/C < 1$ we can bound the $\delta_n^{'}$ and u_n processes for $\parallel \delta_n^{'} + u_n \parallel > C\epsilon$. The pair $(\delta_n^{'}, u_n)$ is then a scale invariant process whose bounded version was proven earlier to converge to zero w.p.1 and therefore by Lemma 2 it too converges to zero w.p.1. This proves the w.p.1 convergence of the triple $\delta_n^{'}$, u_n, and v_n bounding the original process. □

References

Aoki, M. (1967). *Optimization of Stochastic Systems.* New York: Academic Press.

Barto, A. G., Bradtke, S. J., & Singh, S. P. (1993). Learning to act using real-time dynamic programming. Submitted to: *AI Journal.*

Barto, A. G., Sutton, R. S., & Watkins, C.J.C.H. (1990). Sequential decision problems and neural networks. In D. Touretzky (Ed.), *Advances in Neural Information Processing Systems, 2*, pp. 686-693. San Mateo, CA: Morgan Kaufmann.

Bertsekas, D. P. (1987). *Dynamic Programming: Deterministic and Stochastic Models.* Englewood Cliffs, NJ: Prentice-Hall.

Bertsekas, D. P., & Tsitsiklis, J. N. (1989). *Parallel and Distributed Computation: Numerical Methods.* Englewood Cliffs, NJ: Prentice-Hall.

Dayan, P. (1992). The convergence of TD(λ) for general λ. *Machine Learning, 8*, 341-362.

Dayan, P., & Sejnowski, T. J. (1993). *TD(λ) converges with probability 1.* CNL, The Salk Institute, San Diego, CA.

Dvoretzky, A. (1956). On stochastic approximation. *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability.* University of California Press.

Peng J., & Williams R. J. (1993). TD(λ) converges with probability 1. Department of Computer Science preprint, Northeastern University.

Robbins, H., & Monro, S. (1951). A stochastic approximation model. *Annals of Mathematical Statistics, 22*, 400-407.

Ross, S. M. (1970). *Applied Probability Models with Optimization Applications.* San Francisco: Holden-Day.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning, 3*, 9-44.

Tsitsiklis J. N. (1993). Asynchronous stochastic approximation and Q-learning. Submitted to: *Machine Learning.*

Watkins, C.J.C.H. (1989). *Learning from delayed rewards.* PhD Thesis, University of Cambridge, England.

Watkins, C.J.C.H, & Dayan, P. (1992). Q-learning. *Machine Learning, 8*, 279-292.

Werbos, P. (1992). Approximate dynamic programming for real-time control and neural modeling. In D. A. White and D. A. Sofge, (Eds.), *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*, pp. 493-525. New York: Van Nostrand Reinhold.