A Generalized Reinforcement-Learning Model: Convergence and Applicationa

Michael L. Littman[1]
Csaba Szepesva'ri[2]

Technical Report No. CS-96-10

February, 1996

[1]Department of Computer Science, Brown University, 115 Waterman, Providence, RI 02912-1910, mlittman@cs.brown.edu

[2]Bolyal Institute of Mathematics, "Jozsef Attila" University of Szeged, Szeged 6720 Aradi-vrttere 1., HUNGARY, szepes@math.u-szeged.hu

A Generalized Reinforcement-Learning Model: Convergence and Applications

Michael L. Littman
Department of Computer Science
Brown University
Providence, RI 02912-1910
USA
mlittman@cs.brown.edu

Csaba Szepesvári
Bolyai Institute of Mathematics
"József Attila" University of Szeged
Szeged 6720
Aradi vrt tere 1.
HUNGARY
szepes@math.u-szeged.hu

January 18, 1996

Abstract

Reinforcement learning is the process by which an autonomous agent uses its experience interacting with an environment to improve its behavior. The Markov decision process (MDP) model is a popular way of formalizing the reinforcement-learning problem, but it is by no means the only way. In this paper, we show how many of the important theoretical results concerning reinforcement learning in MDPs extend to a generalized MDP model that includes MDPs, two-player games and MDPs under a worst-case optimality criterion as special cases. The basis of this extension is a stochastic-approximation theorem that reduces asynchronous convergence to synchronous convergence.

Keywords: Reinforcement learning, Q-learning convergence, Markov games

1 INTRODUCTION

Reinforcement learning is the process by which an agent improves its behavior in an environment via experience. A *reinforcement-learning scenario* is defined by the experience presented to the agent at each step, and the criterion for evaluating the agent's behavior.

One particularly well-studied reinforcement-learning scenario is that of a single agent maximizing expected discounted total reward in a finite-state environment; experiences are of the form $\langle x, a, y, r \rangle$, where x is a state, a is an action, y is a resulting state and r is the scalar immediate reward to the agent. A discount parameter $0 \leq \gamma < 1$ controls the degree to which future rewards are significant compared to immediate rewards.

The theory of Markov decision processes can be used as a theoretical foundation for important results concerning this reinforcement-learning scenario [1]. A (finite) Markov decision process (MDP) [18] is defined by the tuple $\langle S, A, P, R \rangle$, where S represents a finite set of states, A a finite set of actions, P a transition function, and R a reward function. The optimal behavior for an agent in an MDP depends on the optimality criterion; for the infinite-horizon discounted criterion, the optimal behavior can be found by identifying the optimal value function, defined recursively by

$$V^*(x) = \max_a \left(R(x,a) + \gamma \sum_y P(x,a,y) V^*(y) \right),$$

for all states $x \in S$, where $R(x,a)$ is the immediate reward for taking action a from state x, $0 \leq \gamma < 1$ is a discount factor, and $P(x,a,y)$ is the probability that state y is reached from state x when action $a \in A$ is chosen. These simultaneous equations, known as the *Bellman equations*, can be solved using a variety of techniques ranging from successive approximation [2] to linear programming [6].

In the absence of complete information regarding the transition and reward functions, reinforcement-learning methods can be used to find optimal value functions. Both model-free (direct) methods, such as Q-learning [33], and model-based (indirect) methods, such as prioritized sweeping [15] and DYNA [26], have been explored and many have been shown to converge to optimal value functions under the proper conditions [33, 31, 10, 7].

As we mentioned before, not all reinforcement-learning scenarios of interest can be modeled as MDPs. A great deal of reinforcement-learning research has been directed to the problem of solving two-player games [29, 30, 21, 4], for example, and the reinforcement-learning algorithms for solving MDPs and their convergence proofs do not apply directly to games. In one form of two-player game, experiences are of the form $\langle x, a, y, r \rangle$, where states x and y contain additional information concerning which player (maximizer or minimizer) gets to choose the action in that state, and the optimality criterion is minimax optimality.

There are deep similarities between MDPs and games; for example, it is possible to define a set of Bellman equations for the optimal minimax value of a two-player zero-sum game,

$$V^*(x) = \begin{cases} \max_{a \in A} \left(R(x,a) + \gamma \sum_y P(x,a,y) V^*(y) \right), & \text{if maximizer moves in } x \\ \min_{a \in A} \left(R(x,a) + \gamma \sum_x P(x,a,y) V^*(y) \right), & \text{if minimizer moves in } x, \end{cases}$$

where $R(x,a)$ is the reward to the maximizing player. When $0 \leq \gamma < 1$, these equations have a unique solution and can be solved by successive-approximation methods [23]. In addition, we show in this paper that the natural extension of several reinforcement-learning algorithms for MDPs converge to optimal value functions in two-player games.

In this paper, we introduce a generalized Markov decision process model with applications to reinforcement learning, and list some of the important results concerning the model. Generalized MDPs provide a foundation for the use of reinforcement learning in MDPs and games, as well in risk-sensitive reinforcement learning [8], exploration-sensitive reinforcement learning [11], reinforcement learning in simultaneous-action games [13], and other models. Our main theorem addresses the convergence of asynchronous stochastic processes and shows how this problem can be reduced to determining the convergence of a corresponding synchronous one; it can be used to prove the convergence of model-free and model-based reinforcement-learning algorithms under a variety of different reinforcement-learning scenarios.

In Section 2, we present generalized MDPs, and motivate them using two detailed examples. In Section 3, we describe a stochastic-approximation theorem, and in Section 4 we show several applications of the theorem that prove the convergence of learning processes in generalized MDPs.

2 THE GENERALIZED MODEL

In this section, we introduce our generalized MDP model. We begin by summarizing some of the more significant results regarding the standard MDP model and some important results for two-player games.

2.1 MARKOV DECISION PROCESSES

To provide a point of departure for our generalization of Markov decision processes, we begin by describing some results concerning the use of reinforcement learning in the MDP scenario described earlier. These results are well established; proofs of the unattributed claims can be found in Puterman's MDP book [18].

The ultimate target of learning is an optimal policy. A *policy* is some function that tells the agent which actions should be chosen under which circumstances. A policy π is *optimal* under the expected discounted total reward criterion if, with respect to the space of all possible policies, π maximizes the expected discounted total reward from all states.

Maximizing over the space of all possible policies is practically infeasible. However, MDPs have an important property that makes it unnecessary to consider such a broad space of possibilities. We say a policy π is *stationary* and *deterministic* if it maps directly from states to actions, ignoring everything else, and we write $\pi(x)$ as the action chosen by π when the current state is x. In expected discounted total reward MDP environments, there is always a stationary deterministic policy that is optimal; we will therefore use the word "policy" to mean stationary deterministic policy, unless otherwise stated.

The value function for a policy π, V^π, maps states to their expected discounted total reward under policy π. It can be defined by the simultaneous equations

$$V^\pi(x) = R(x,a) + \gamma \sum_y P(x,a,y) V^\pi(y).$$

It is also possible to condition the immedate rewards on the state y as well; this is somewhat more general, but complicates the presentation. The optimal value function V^* is the value function of an optimal policy; it is unique for $0 \leq \gamma < 1$. The *myopic policy* with respect to a value function V is the policy π_V such that

$$\pi_V(x) = \arg\max_a \left(R(x,a) + \gamma \sum_y P(x,a,y)V(y) \right).$$

Any myopic policy with respect to the optimal value function is optimal.

The Bellman equations can be operationalized in the form of the dynamic-programming operator T, which maps value functions to value functions:

$$[TV](x) = \max_a \left(R(x,a) + \gamma \sum_y P(x,a,y)V(y) \right).$$

For $0 \leq \gamma < 1$, successive applications of T to a value function bring it closer and closer to the optimal value function V^*, which is the unique fixed point of T: $V^* = TV^*$.

In reinforcement learning, R and P are not known in advance. They can be learned from experience by keeping statistics on the expected reward for each state-action pair, and the proportion of transitions to each next state for each state-action pair. In model-based reinforcement learning, R and P are estimated on-line, and the value function is updated according to the approximate dynamic-programming operator derived from these estimates; this algorithm converges to the optimal value function under a wide variety of choices of the order states are updated [7].

The method of Q-learning [32] uses experience to estimate the optimal value function without ever explicitly approximating R and P. The algorithm estimates the optimal Q function

$$Q^*(x,a) = R(x,a) + \gamma \sum_y P(x,a,y)V^*(y),$$

from which the optimal value function can be computed via $V^*(x) = \max_a Q^*(x,a)$. Given the experience at step t $\langle x_t, a_t, y_t, r_t \rangle$ and the current estimate $Q_t(x,a)$ of the optimal Q function, Q-learning updates

$$Q_{t+1}(x_t,a_t) := (1 - \alpha_t(x_t,a_t))Q_t(x_t,a_t) + \alpha_t(x_t,a_t)(r_t + \gamma \max_a Q_t(y_t,a)),$$

where $0 \leq \alpha_t(x,a) \leq 1$ is a learning rate that controls how quickly new estimates are blended into old estimates as a function of the state-action pair and the trial number. Q-learning converges to the optimal Q function under the proper conditions [33, 31, 10].

2.2 ALTERNATING MARKOV GAMES

In alternating Markov games, two players take turns issuing actions to try to maximize their own expected discounted total reward. The model is defined by the tuple $\langle S_1, S_2, A, B, P, R \rangle$, where S_1 is the set of states in which player 1 issues actions from the set A, S_2 is the set of states in which player 2 issues actions from the set B, P is the transition function, and R is the reward function for player 1. In the zero-sum games we consider, the rewards to player 2 (the minimizer) are simply the additive inverse of the rewards for player 1 (the maximizer). Markov decision processes are a special case of alternating Markov games in which $S_2 = \emptyset$; Condon [5] proves this and the other unattributed results in this section.

A common optimality criterion for alternating Markov games is discounted minimax optimality. Under this criterion, the maximizer should choose actions so as to maximize its reward in the event that the minimizer chooses the best possible counter-policy. An equivalent definition is for the minimizer to choose actions to minimize its reward against the maximizer with the best possible counter-policy. A pair of policies is said to be in *equilibrium* if neither player has any incentive to change policies if the other player's policy remains fixed. The value function for a pair of equilibrium policies is the optimal value function for the game; it is unique when $0 \le \gamma < 1$, and can be found by successive approximation. For both players, there is always a deterministic stationary optimal policy. Any myopic policy with respect to the optimal value function is optimal, and any pair of optimal policies is in equilibrium.

Dynamic-programming operators, Bellman equations, and reinforcement-learning algorithms can be defined for alternating Markov games by starting with the definitions used in MDPs and changing the maximum operators to either maximums or minimums conditioned on the state. We show below that the resulting algorithms share their convergence properties with the analogous algorithms for MDPs.

2.3 GENERALIZED MDPS

In alternating Markov games and MDPs, optimal behavior can be specified by the Bellman equations; any myopic policy with respect to the optimal value function is optimal. In this section, we generalize the Bellman equations to define optimal behavior for a broad class of reinforcement-learning models. The objective criterion used in these models is additive in that the the value of a policy is some measure of the *total* reward received.

The generalized Bellman equations can be written

$$V^*(x) = \bigotimes_a \left(R(x,a) + \gamma \bigoplus_y V^*(y) \right). \tag{1}$$

Here, "$\otimes$" and "$\oplus$" represent operators that summarize values over actions as a function of the state x and next states as a function of the state-action pair (x,a), respectively. For Markov decision processes, $\bigotimes_a f(x,a) = \max_a f(x,a)$ and $\bigoplus_y g(x,a,y) = \sum_y P(x,a,y) g(x,a,y)$. For alternating Markov games, $\bigoplus_y g(x,a,y) = \sum_y P(x,a,y) g(x,a,y)$ and $\bigotimes_a f(x,a) = \max_a f(x,a)$ or $\min_a f(x,a)$ depending whether x is in S_1 or S_2. A large variety of other models can be represented in this framework; several examples are discussed in Section 4.

From a reinforcement-learning perspective, the value functions defined by the generalized MDP model can be interpreted as the total value of the rewards received by an agent selecting actions in a stochastic environment. The agent begins in state x, takes action a, and ends up in state y. The $\oplus$ operator defines how the value of the next state should be used in assigning value to the current state. The $\otimes$ operator defines how an optimal agent should choose actions.

When $0 \le \gamma < 1$ and $\otimes$ and $\oplus$ are non-expansions, the generalized Bellman equations have a unique optimal solution, and therefore, the optimal value function is well defined.

model/example reference	$\bigotimes_a f(x,a)$	$\bigoplus_y g(x,a,y)$
disc. exp. MDPs [33]	$\max_a f(x,a)$	$\sum_y P(x,a,y)g(x,a,y)$
exp. return of π [25]	$\sum_a \pi(x,a)f(x,a)$	$\sum_y P(x,a,y)g(x,a,y)$
alt. Markov games [4]	$\max_a$ or $\min_a f(x,a)$	$\sum_y P(x,a,y)g(x,a,y)$
risk-sensitive MDPs [8]	$\max_a f(x,a)$	$\min_{y:P(x,a,y)>0} g(x,a,y)$
exploration-sens. MDPs [11]	$\max_{\pi \in P_0} \sum_a \pi(x,a)f(x,a)$	$\sum_y P(x,a,y)g(x,a,y)$
Markov games [13]	$\max_A \min_b \sum_a A(a)f(x,(a,b))$	$\sum_y P(x,(a,b),y)g(x,(a,b),y)$
information-state MDP [16]	$\max_a f(x,a)$	$\sum_{y \in N(x,a)} P(x,a,y)g(x,a,y)$

Table 1: Some reinforcement-learning scenarios and their specification as generalized Markov decision processes.

The $\otimes$ operator is a non-expansion if

$$\left| \bigotimes_a f_1(x,a) - \bigotimes_a f_2(x,a) \right| \le \max_a |f_1(x,a) - f_2(x,a)|$$

for all f_1, f_2, and x. An analogous condition defines when $\oplus$ is a non-expansion.

Many natural operators are non-expansions, such as max, min, midpoint, median, mean, and fixed weighted averages of these operations. Mode and Boltzmann-weighted averages are not non-expansions. Several previously described reinforcement-learning scenarios are special cases of this generalized MDP model—Table 1 gives a brief sampling. For more information about the specific models listed, see the associated references.

As with MDPs, we can define a dynamic-programming operator

$$[TV](x) = \bigotimes_a \left(R(x,a) + \gamma \bigoplus_y V(y) \right) \tag{2}$$

such that for $0 \le \gamma < 1$ the optimal value function V^* is the unique fixed point of T. The operator T is a contraction mapping as long as $\gamma < 1$. Recall that an operator T is a contraction mapping if

$$\sup_x |[TV_1](x) - [TV_2](x)| \le \gamma \sup_x |V_1(x) - V_2(x)|$$

where V_1 and V_2 are arbitrary functions and $0 \le \gamma < 1$ is the index of contraction.

We can define a notion of stationary myopic policies with respect to a value function V; it is any (stochastic) policy π_V for which $T^\pi V = TV$ where

$$[T^\pi V](x) = \sum_a \pi(x,a) \left(R(x,a) + \gamma \bigoplus_y V(y) \right).$$

Here $\pi(x,a)$ represents the probability that an agent following π would choose action a in state x. To be certain that every value function possesses a myopic policy, we require that the operator $\otimes$ satisfy the following property: for all functions f and states x, $\min_a f(x,a) \le \bigotimes_a f(x,a) \le \max_a f(x,a)$.

The value function with respect to a policy π, V^π can be defined by the simultaneous equations

$$V^\pi(x) = \sum_a \pi(x,a) \left(R(x,a) + \gamma \bigoplus_y V^\pi(y) \right);$$

it is unique. A policy π is optimal if it is myopic with respect to its own value function. If π^* is an optimal policy, then V^{π^*} is the fixed point of T because $V^{\pi^*} = T^{\pi^*} V^{\pi^*} = TV^{\pi^*}$. Thus, $V^{\pi^*} = V^*$, when $\gamma < 1$ because T has a unique fixed point.

The next section describes a general theorem that can be used to prove the convergence of several reinforcement-learning algorithms for these and other models.

3 CONVERGENCE THEOREM

The process of finding an optimal value function can be viewed in the following general way. At any moment in time, there is a set of values representing the current approximation of the optimal value function. On each iteration, we apply some dynamic-programming operator, perhaps modified by experience, to the current approximation to generate a new approximation. Over time, we would like the approximation to tend toward the optimal value function.

In this process, there are two types of approximation going on simultaneously. The first is an approximation of the dynamic-programming operator for the underlying model, and the second is the use of the approximate dynamic-programming operator to find the optimal value function. This section presents a theorem that gives a set of conditions under which this type of simultaneous stochastic approximation converges to an optimal value function.

First, we need to define the general stochastic process. Let the set X be the states of the model, and the set $\mathbb{B}(X)$ of bounded, real-valued functions over X be the set of value functions. Let $T : \mathbb{B}(X) \to \mathbb{B}(X)$ be an arbitrary contraction mapping and V^* be the fixed point of T.

If we had direct access to the contraction mapping T, we could use it to successively approximate V^*. In most reinforcement-learning scenarios, T is not available and we must use our experience to construct approximations of T. Consider a sequence of random operators $T_t : \mathbb{B}(X) \to (\mathbb{B}(X) \to \mathbb{B}(X))$ and define $U_{t+1} = [T_t U_t]V$ where V and $U_0 \in \mathbb{B}(X)$ are arbitrary value functions. We say T_t approximates T at V with probability 1 uniformly over X, if U_t converges to TV uniformly over X[1]. The basic idea is that T_t is a randomized version of T in some sense; it uses U_t as "memory" to help it approximate TV.

The following theorem shows that, under the proper conditions, we can use the sequence T_t to estimate the fixed point V^* of T.

Theorem 1 *Let T be an arbitrary mapping with fixed point V^*, and let T_t approximate T at V^* with probability 1 uniformly over X. Let V_0 be an arbitrary value function, and define $V_{t+1} = [T_t V_t]V_t$. If there exist functions $0 \le F_t(x) \le 1$ and $0 \le G_t(x) \le 1$ satisfying the conditions below with probability one, then V_t converges to V^* with probability 1 uniformly over X:*

[1]A sequence of functions f_n converges to f^* with probability 1 uniformly over X if, for the events w for which $f_n(w,x) \to f^*$, the convergence is uniform in x.

1. *for all U_1, and $U_2 \in \mathbb{B}(X)$ and all $x \in X$,*

$$|([T_tU_1]V^*)(x) - ([T_tU_2]V^*)(x)| \leq G_t(x) \sup_{x'} |U_1(x') - U_2(x')|;$$

2. *for all U and $V \in \mathbb{B}(X)$, and all $x \in X$,*

$$|([T_tU]V^*)(x) - ([T_tU]V)(x)| \leq F_t(x) \sup_{x'} |V^*(x') - V(x')|;$$

3. *for all $k > 0$, $\Pi_{t=k}^n G_t(x)$ converges to zero uniformly in x as n increases; and,*

4. *there exists $0 \leq \gamma < 1$ such that for all $x \in X$ and large enough t,*

$$F_t(x) \leq \gamma(1 - G_t(x)).$$

Note that from the conditions of the theorem, it follows that T is a contraction operator at V^* with index of contraction γ. The theorem is proven in an extended version of this paper [28]. We next describe some of the intuition behind the statement of the theorem and its conditions.

The iterative approximation of V^* is performed by computing $V_{t+1} = [T_tV_t]V_t$, where T_t approximates T with the help of the "memory" present in V_t. Because of Conditions 1 and 2, $G_t(x)$ is the extent to which the estimated value function depends on its present value and $F_t(x) \approx 1 - G_t(x)$ is the extent to which the estimated value function is based on "new" information (this reasoning becomes clearer in the context of the applications in Section 4).

In some applications, such as Q-learning, the contribution of new information needs to decay over time to insure that the process converges. In this case, $G_t(x)$ needs to converge to one. Condition 3 allows $G_t(x)$ to converge to 1 as long as the convergence is slow enough to incorporate sufficient information for the process to converge.

Condition 4 links the values of $G_t(x)$ and $F_t(x)$ through some quantity $\gamma < 1$. If it were somehow possible to update the values synchronously over the entire state space, the process would converge to V^* even when $\gamma = 1$. In the more interesting asynchronous case, when $\gamma = 1$, the long-term behavior of V_t is not immediately clear; it may even be that V_t converges to something other than V^*. The requirement that $\gamma < 1$ insures that the use of outdated information in the asynchronous updates does not cause a problem in convergence.

One of the most noteworthy aspects of this theorem is that it shows how to reduce the problem of approximating V^* to the problem of approximating T at a particular point V (in particular, it is enough if T can be approximated at V^*); in many cases, the latter is much easier to achieve and also to prove. For example, the theorem makes the convergence of Q-learning a consequence of the classical Robbins-Monro theorem [20].

4 APPLICATIONS

This section makes use of Theorem 1 to prove the convergence of various reinforcement-learning algorithms.

4.1 GENERALIZED Q-LEARNING FOR EXPECTED VALUE MODELS

Consider the family of finite state and action generalized MDPs defined by the Bellman equations

$$V^*(x) = \bigotimes_a \left(R(x,a) + \gamma \sum_y P(x,a,y) V^*(y) \right)$$

where the definition of $\otimes$ does not depend on R or P. A Q-learning algorithm for this class of models can be defined as follows. Given experience $\langle x_t, a_t, y_t, r_t \rangle$ at time t and an estimate $Q_t(x,a)$ of the optimal Q function, let

$$Q_{t+1}(x_t, a_t) := (1 - \alpha_t(x_t, a_t)) Q_t(x_t, a_t) + \alpha_t(x_t, a_t) \left(r_t + \gamma \bigotimes_a Q_t(y_t, a) \right).$$

We can derive the assumptions necessary for this learning algorithm to satisfy the conditions of Theorem 1 and therefore converge to the optimal Q values. The dynamic-programming operator defining the optimal Q function is

$$[TQ](x,a) = R(x,a) + \gamma \sum_y P(x,a,y) \bigotimes_{a'} Q(y,a').$$

The randomized approximate dynamic-programming operator that gives rise to the Q-learning rule is

$$([T_t Q']Q)(x,a) = \begin{cases} (1-\alpha_t(x,a))Q'(x,a) + \alpha_t(x,a)(r_t + \gamma \otimes_{a'} Q(y_t, a')), & \text{if } x = x_t \text{ and } a = a_t \\ Q'(x,a), & \text{otherwise.} \end{cases}$$

If

- y_t is randomly selected according to the probability distribution defined by $P(x_t, a_t, \cdot)$,
- $\otimes$ is a non-expansion, and both the expected value and the variance of $\otimes_a Q(y_t, a)$ exist given the way y_t is sampled,
- r_t has finite variance and expected value given x_t and a_t equal to $R(x_t, a_t)$,
- the learning rates are decayed so that $\sum_t \chi(x_t = x, a_t = a)\alpha_t(x,a) = \infty$ and $\sum_t \chi(x_t = x, a_t = a)\alpha_t(x,a)^2 < \infty$ with probability 1 uniformly over $X \times A$ [2],

then a standard result from the theory of stochastic approximation [20] states that T_t approximates T with probability 1 uniformly over $X \times A$. That is, this method of using a decayed, exponentially weighted average correctly computes the average one-step reward.

Let

$$G_t(x,a) = \begin{cases} 1 - \alpha_t(x,a), & \text{if } x = x_t \text{ and } a = a_t; \\ 0, & \text{otherwise,} \end{cases}$$

[2]This condition implies, among other things, that every state-action pair is updated infinitely often. Here, χ denotes the characteristic function.

and

$$F_t(x,a) = \begin{cases} \gamma\alpha_t(x,a), & \text{if } x = x_t \text{ and } a = a_t; \\ 0, & \text{otherwise.} \end{cases}$$

These functions satisfy the conditions of Theorem 1 (Condition 3 is implied by the restrictions placed on the sequence of learning rates α_t).

Theorem 1 therefore implies that this generalized Q-learning algorithm converges to the optimal Q function with probability 1 uniformly over $X \times A$. The convergence of Q-learning for discounted MDPs and alternating Markov games follows trivially from this. Extensions of this result for undiscounted "all-policies-proper" MDPs [3], a soft state aggregation learning rule [24], and a "spreading" learning rule [19] are given in an extended version of this paper [28].

4.2 Q-LEARNING FOR MARKOV GAMES

Markov games are a generalization of MDPs and alternating Markov games in which both players simultaneously choose actions at each step. The basic model was developed by Shapley [23] and is defined by the tuple $\langle S, A, B, P, R\rangle$ and discount factor γ. As in alternating Markov games, the optimality criterion is one of discounted minimax optimality, but because the players move simultaneously, the Bellman equations take on a more complex form:

$$V^*(x) = \max_{\rho\in\Pi(A)} \min_{b\in B} \sum_{a\in A} \rho(a) \left(R(x,a,b) + \gamma \sum_{y\in S} P(x,a,b,y) V^*(y) \right).$$

In these equations, $R(x,a,b)$ is the immediate reward for the maximizer for taking action a in state x at the same time the minimizer takes action b, $P(x,a,b,y)$ is the probability that state y is reached from state x when the maximizer takes action a and the minimizer takes action b, and $\Pi(A)$ represents the set of discrete probability distributions over the set A. The sets S, A, and B are finite.

Once again, optimal policies are policies that are in equilibrium, and there is always a pair of optimal policies that are stationary. Unlike MDPs and alternating Markov games, the optimal policies are sometimes stochastic; there are Markov games in which no deterministic policy is optimal. The stochastic nature of optimal policies explains the need for the optimization over probability distributions in the Bellman equations, and stems from the fact that players must avoid being "second guessed" during action selection. An equivalent set of equations can be written with a stochastic choice for the minimizer, and also with the roles of the maximizer and minimizer reversed.

The Q-learning update rule for Markov games [13] given step t experience $\langle x_t, a_t, b_t, y_t, r_t\rangle$ has the form

$$Q_{t+1}(x_t,a_t,b_t) := (1-\alpha_t(x_t,a_t,b_t))Q_t(x_t,a_t,b_t) + \alpha_t(x_t,a_t,b_t)\left(r_t + \gamma \bigotimes_{a,b} Q_t(y_t,a,b) \right),$$

where

$$\bigotimes_{a,b} g(x,a,b) = \max_{\rho\in\Pi(A)} \min_{b\in B} \sum_{a\in A} \rho(a) g(x,a,b).$$

The results of the previous section prove that this rule converges to the optimal Q function under the proper conditions.

4.3 RISK-SENSITIVE MODELS

Heger [8] described an optimality criterion for MDPs in which only the *worst* possible value of the next state makes a contribution to the value of a state. An optimal policy under this criterion is one that avoids states for which a bad outcome is possible, even if it is not probable; for this reason, the criterion has a risk-averse quality to it. The generalized Bellman equations for this criterion are

$$V^*(x) = \bigotimes_a \left(R(x,a) + \gamma \min_{y:P(x,a,y)>0} V^*(y) \right).$$

The argument in Section 4.5 shows that model-based reinforcement learning can be used to find optimal policies in risk-sensitive models, as long as $\otimes$ does not depend on R or P, and P is estimated in a way that preserves its zero vs. non-zero nature in the limit.

For the model in which $\bigotimes_a f(x,a) = \max_a f(x,a)$, Heger defined a Q-learning-like algorithm that converges to optimal policies without estimating R and P online. In essence, the learning algorithm uses an update rule analogous to the rule in Q-learning with the additional requirement that the initial Q function be set optimistically; that is, $Q_0(x,a)$ must be larger than $Q^*(x,a)$ for all x and a. Like Q-learning, this learning algorithm is a generalization of Korf's [12] LRTA* algorithm for stochastic environments.

Using Theorem 1 it is possible to prove the convergence of a generalization of Heger's algorithm to models where $\bigotimes_a f(x,a) = f(x, a^*(f,x))$ for some function $a^*(\cdot)$; that is, as long as the summary value of $f(x,a)$ is equal to $f(x,a^*)$ for some a^*. The proof is based on estimating the Q-learning algorithm from above by an appropriate process where the Q function is updated only if the received experience tuple is an extremity according to the optimality equation; details are given in the extended paper [28].

4.4 EXPLORATION-SENSITIVE MODELS

John [11] considered the implications of insisting that reinforcement-learning agents keep exploring forever; he found that better learning performance can be achieved if the Q-learning rule is changed to incorporate the condition of persistent exploration. In John's formulation, the agent is forced to adopt a policy from a restricted set; in one example, the agent must choose a stochastic stationary policy that selects actions at random 5% of the time.

This approach requires that the definition of optimality be changed to reflect the restriction on policies. The optimal value function is given by $V^*(x) = \sup_{\pi \in P_0} V^\pi(x)$, where P_0 is the set of permitted (stationary) policies, and the associated Bellman equations are

$$V^*(x) = \sup_{\pi \in P_0} \sum_a \pi(x,a) \left(R(x,a) + \gamma \sum_y P(x,a,y) V^*(y) \right),$$

which corresponds to a generalized MDP model with $\bigoplus_y g(x,a,y) = \sum_y P(x,a,y) g(x,a,y)$ and $\bigotimes_a f(x,a) = \sup_{\pi \in P_0} \sum_a \pi(x,a) f(x,a)$. Because $\pi(x,\cdot)$ is a probability distribution for any given state x, $\otimes$ is a non-expansion and, thus, the convergence of the associated Q-learning algorithm follows from the arguments in Section 4.1. As a result, John's learning rule gives the optimal policy under the revised optimality criterion.

4.5 MODEL-BASED METHODS

The defining assumption in reinforcement learning is that the reward and transition functions, R and P, are not known in advance. Although Q-learning shows that optimal value functions can be estimated without ever explicitly learning R and P, learning R and P makes more efficient use of experience at the expense of additional storage and computation [15]. The parameters of R and P can be learned from experience by keeping statistics for each state-action pair on the expected reward and the proportion of transitions to each next state. In model-based reinforcement learning, R and P are estimated on-line, and the value function is updated according to the approximate dynamic-programming operator derived from these estimates. Theorem 1 implies the convergence of a wide variety of model-based reinforcement-learning methods.

The dynamic-programming operator defining the optimal value for generalized MDPs is given in Equation 2. Here we assume that $\bigoplus$ may depend on P and/or R, but $\bigotimes$ may not. It is possible to extend the following argument to allow $\bigotimes$ to depend on P and R as well. In model-based reinforcement learning, R and P are estimated by the quantities R_t and P_t, and $\bigoplus^t$ is an estimate of the $\bigoplus$ operator defined using R_t and P_t. As long as every state-action pair is visited infinitely often, there are a number of simple methods for computing R_t and P_t that converge to R and P. A bit more care is needed to insure that $\bigoplus^t$ converges to $\bigoplus$, however. For example, in expected-reward models, $\bigoplus_y g(x,a,y) = \sum_y P(x,a,y)g(x,a,y)$ and the convergence of P_t to P guarantees the convergence of $\bigoplus^t$ to $\bigoplus$. On the other hand, in a risk-sensitive model, $\bigoplus_y g(x,a,y) = \min_{y:P(x,a,y)>0} g(x,a,y)$ and it is necessary to approximate P in a way that insures that the set of y such that $P_t(x,a,y) > 0$ converges to the set of y such that $P(x,a,y) > 0$. This can be accomplished easily, for example, by setting $P_t(x,a,y) = 0$ if no transition from x to y under a has been observed.

Assuming P and R can be estimated in a way that results in the convergence of $\bigoplus^t$ to $\bigoplus$, the approximate dynamic-programming operator T_t defined by

$$([T_tU]V)(x) = \begin{cases} \bigotimes_a \left(R_t(x,a) + \gamma \bigoplus_y^t V(y)\right), & \text{if } x \in \tau_t \\ U(x), & \text{otherwise,} \end{cases}$$

converges to T with probability 1 uniformly. Here, the set $\tau_t \subseteq S$ represents the set of states whose values are updated on step t; one popular choice is to set $\tau_t = \{x_t\}$.

The functions

$$G_t(x) = \begin{cases} 0, & \text{if } x \in \tau_t; \\ 1, & \text{otherwise,} \end{cases}$$

and

$$F_t(x) = \begin{cases} \gamma, & \text{if } x \in \tau_t; \\ 0, & \text{otherwise,} \end{cases}$$

satisfy the conditions of Theorem 1 as long as each x is in infinitely many τ_t sets (Condition 3) and the discount factor γ is less than 1 (Condition 4).

As a consequence of this argument and Theorem 1, model-based methods can be used to find optimal policies in MDPs, alternating Markov games, Markov games, risk-sensitive MDPs, and exploration-sensitive MDPs. Also, if $R_t = R$ and $P_t = P$ for all t, this result implies that real-time dynamic programming converges to the optimal value function [1].

5 CONCLUSIONS

In this paper, we presented a generalized model of Markov decision processes, and proved the convergence of several reinforcement-learning algorithms in the generalized model.

Other Results We have derived a collection of results [28] for the generalized MDP model that demonstrate its general applicability: the Bellman equations can be solved by value iteration; a myopic policy with respect to an approximately optimal value function gives an approximately optimal policy [34, 9]; when $\bigotimes$ has a particular "maximization" property, policy iteration converges to the optimal value function; and, for models with finite state and action spaces, both value iteration and policy iteration identify optimal policies in pseudopolynomial time.

Related Work The work presented here is closely related to several previous research efforts. Szepesvári [27] described a related generalized reinforcement-learning model, and presented conditions under which there is an optimal (stationary) policy that is myopic with respect to the optimal value function.

Jaakkola, Jordan, and Singh [10] and Tsitsiklis [31] developed the connection between stochastic-approximation theory and reinforcement learning in MDPs. Our work is similar in spirit to that of Jaakkola, et al. We believe the form of Theorem 1 makes it particularly convenient for proving the convergence of reinforcement-learning algorithms; our theorem reduces the proof of the convergence of an asynchronous process to a simpler proof of convergence of a corresponding synchronized one. This idea enables us to prove the convergence of asynchronous stochastic processes whose underlying synchronous process is not of the Robbins-Monro type (e.g., risk-sensitive MDPs, model-based algorithms, etc.).

Future Work There are many areas of interest in the theory of reinforcement learning that we would like to address in future work. The results in this paper primarily concern reinforcement-learning in contractive models ($\gamma < 1$ or all-policies-proper), and there are important non-contractive reinforcement-learning scenarios, for example, reinforcement learning under an average-reward criterion [22, 14]. It would be interesting to develop a TD(λ) algorithm [25] for generalized MDPs; this has already been done for MDPs [17]. Theorem 1 is not restricted to finite state spaces, and it might be valuable to prove the convergence of a reinforcement-learning algorithm for a infinite state-space model.

Conclusion By identifying common elements among several reinforcement-learning scenarios, we created a new class of models that generalizes existing models in an interesting way. In the generalized framework, we replicated the established convergence proofs for reinforcement learning in Markov decision processes, and proved new results concerning the convergence of reinforcement-learning algorithms in game environments, under a risk-sensitive assumption, and under an exploration-sensitive assumption. At the heart of our results is a new stochastic-approximation theorem that is easy to apply to new situations.

References

[1] Andrew G. Barto, Richard S. Sutton, and Christopher J. C. H. Watkins. Learning and sequential decision making. Technical Report 89-95, Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts, 1989. Also published in *Learning and Computational Neuroscience: Foundations of Adaptive Networks*, Michael Gabriel and John Moore, editors. The MIT Press, Cambridge, Massachusetts, 1991.

[2] Richard Bellman. *Dynamic Programming.* Princeton University Press, Princeton, NJ, 1957.

[3] Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods.* Prentice-Hall, Englewood Cliffs, NJ, 1989.

[4] Justin A. Boyan. Modular neural networks for learning context-dependent game strategies. Master's thesis, Department of Engineering and Computer Laboratory, University of Cambridge, Cambridge, UK, August 1992.

[5] Anne Condon. The complexity of stochastic games. *Information and Computation*, 96(2):203–224, February 1992.

[6] Cyrus Derman. *Finite State Markovian Decision Processes.* Academic Press, New York, 1970. Volume 67 is Mathematics in Science and Engineering, edited by Richard Bellman.

[7] Vijaykumar Gullapalli and Andrew G. Barto. Convergence of indirect adaptive asynchronous value iteration algorithms. In S. J. Hanson, J. D Cowan, and C. L. Giles, editors, *Advances in Neural Information Processing Systems 6*, pages 695–702. Morgan Kaufmann, April 1994.

[8] Matthias Heger. Consideration of risk in reinforcement learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 105–111, San Francisco, CA, 1994. Morgan Kaufmann.

[9] Matthias Heger. The loss from imperfect value functions in expectation-based and minimax-based tasks. *Machine Learning*, 1995. In preparation.

[10] Tommi Jaakkola, Michael I. Jordan, and Satinder P. Singh. On the convergence of stochastic iterative dynamic programming algorithms. *Neural Computation*, 6(6), November 1994.

[11] George H. John. When the best move isn't optimal: Q-learning with exploration. Technical report, Stanford University, 1995. Available on the web.

[12] R. E. Korf. Real-time heuristic search. *Artificial Intelligence*, 42:189–211, 1990.

[13] Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 157–163, San Francisco, CA, 1994. Morgan Kaufmann.

[14] Sridhar Mahadevan. Average reward reinforcement learning: Foundations, algorithms, and empirical results. *Machine Learning*, to appear.

[15] Andrew W. Moore and Christopher G. Atkeson. Prioritized sweeping: Reinforcement learning with less data and less real time. *Machine Learning*, 13, 1993.

[16] Ronald Parr and Stuart Russell. Approximating optimal policies for partially observable stochastic domains. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1995.

[17] Jing Peng and Ronald J. Williams. Incremental multi-step Q-learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 226–232, San Francisco, CA, 1994. Morgan Kaufmann.

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

[19] Carlos Ribeiro and Csaba Szepesvári. Q-learning with a spreading activiation rule. Submitted to ML'96, 1996.

[20] H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

[21] Nicol N. Schraudolph, Peter Dayan, and Terrence J. Sejnowski. Using the TD(λ) algorithm to learn an evaluation function for the game of Go. In *Advances in Neural Information Processing Systems 6*, San Mateo, CA, 1994. Morgan Kaufmann.

[22] Anton Schwartz. A reinforcement learning method for maximizing undiscounted rewards. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 298–305, Amherst, Massachusetts, 1993. Morgan Kaufmann.

[23] L.S. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences of the United States of America*, 39:1095–1100, 1953.

[24] S.P. Singh, T. Jaakkola, and M.I. Jordan. Reinforcement learning with soft state aggregation. In *Proceedings of Neural Information Processing Systems*, 1995.

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

[26] Richard S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proceedings of the Seventh International Conference on Machine Learning*, Austin, TX, 1990. Morgan Kaufmann.

[27] Csaba Szepesvári. General framework for reinforcement learning. In *Proceedings of ICANN'95 Paris*, 1995.

[28] Csaba Szepesvári and Michael L. Littman. Generalized markov decision processes: Dynamic-programming and reinforcement-learning algorithms. In preparation, 1996.

[29] Gerald Tesauro. Temporal difference learning and TD-Gammon. *Communications of the ACM*, pages 58–67, March 1995.

[30] Sebastian Thrun. Learning to play the game of chess. In *Neural Information Processing Systems 7*, 1995.

[31] John N. Tsitsiklis. Asynchronous stochastic approximation and Q-learning. *Machine Learning*, 16(3), September 1994.

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

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

[34] Ronald J. Williams and Leemon C. Baird, III. 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 1993.