Convergent Reinforcement Learning with Value Function Interpolation

Csaba Szepesvári
Mindmaker Ltd., Budapest 1121, Konkoly Thege M. u. 29–33, HUNGARY

SZEPES@MINDMAKER.HU

Abstract
We consider the convergence of a class of reinforcement learning algorithms combined with value function interpolation methods using the methods developed in (Littman & Szepesvári, 1996). As a special case of the obtained general results, for the first time, we prove the (almost sure) convergence of Q-learning when combined with value function interpolation in uncountable spaces.

1. Introduction
Since the early days of dynamic programming researchers interested in solving problems with large or even infinite state spaces combined function approximators and value backups. Some of the first examples include (Bellman & Dreyfus, 1959; Reetz, 1977; Werbo, 1977). The approach at these days was to use some interpolation method and compute the value backups on a set of representative states.

Reinforcement learning (RL) has grown out of the work of (Samuel, 1959) who himself has used a form of value function approximation. Sometimes RL is defined as large-scale approximate dynamic programming combined with learning techniques.

One main stream of RL employs gradient-based methods. Focusing on value function approximation only, examples include TD(λ) (Sutton, 1988), Bellman error (Schweitzer & Seidman, 1985) and residual algorithms (Baird, 1995; Baird & Moore, 1999). Some successful applications of these methods are described in (Tesauro, 1994; Zhang & Dietterich, 1995; Baxter et al., 1998). Another main stream employs constructive (local) methods. Decision tree based methods were used e.g. in (Chapman & Kaelbling, 1991) and more recently in (Wang & Dietterich, 1999). Discretization using triangularization has been advocated in (Munce, 1997), (sometimes adaptive) state aggregation was employed in e.g. (Moore & Atkeson, 1995; Tsitsiklis & Van Roy, 1996; Gordon, 1995). Recently, locally weighted regression models have been used by (Smart & Kaelbling, 2000).

Regression literature teaches us that the advantage of constructive or local methods is that the behavior of the method can be understood much better than that of the gradient-based approaches. As discussed in (Atkeson et al., 1997) advantages of constructive local methods include (i) no interference, (ii) incremental improvement (no local minima as in gradient methods), (iii) one-shot learning and (iv) that the incorporation of new data is cheap. The main disadvantage of local methods is that value prediction slows down as the number of training examples increases and their performance degrades as the dimensionality of the space increases.

Unfortunately, it seems that the nice properties of constructive approximations do not carry over seamlessly to the combinations of local methods and reinforcement learning. This has been observed by many researchers (e.g. (Boyan & Moore, 1995; Baird, 1995)) and then researchers were urged to come up with convergent algorithms. In (Gordon, 1995) and independently in (Tsitsiklis & Van Roy, 1996) convergence results were derived for approximate dynamic programming when the “value-fitting operator” was chosen to be a non-expansion. More precisely, in (Gordon, 1995) an algorithm of the form \( V_{i+1} = TV_i \) was considered, where \( T \) is the value backup operator and \( \mathcal{G} \) is a non-expansion w.r.t. the supremum norm. In discounted problems \( T \) is known to be a contraction (w.r.t. the supremum norm) and thus \( T = GT \) is a contraction as well and the algorithm converges to the unique fixed point of \( T \). In order to make the algorithm practical one decomposes \( \mathcal{G} \) into the product of an operator \( \mathcal{P} \) that maps the space of value functions into a low-dimensional space \( \Theta \) and an operator \( \mathcal{E} \) that maps \( \Theta \) to the space of value functions. Then, by defining

\[
\theta_{i+1} = \mathcal{P} \mathcal{T} \mathcal{E} \theta_i, \tag{1}
\]

\( \theta_{i+1} = \mathcal{P} \mathcal{T} \mathcal{E} \theta_i, \)
Gordon proposed to use “averagers” as $\mathcal{E}$ and discussed the problems with the extension of the method to the learning scenario when $T$ is unknown.

In (Tsitsiklis & Van Roy, 1996), $T$ is still assumed to be known, although these authors mention that their method could be extended to the case when $T$ is replaced by its Monte-Carlo approximation. State aggregation with approximate gradual value iteration is studied. Iteration $i$ is replaced by $\theta_{i+1} = (1-\alpha_i)\theta_i + \alpha_i(T(\mathcal{E}\theta_i))(X_i^{[i]}),$ where $X_i^{[i]}$ is a sample from the $i$th “cluster” and $(\mathcal{E}\theta_i)(x) = \sum_{i=1}^{n} \theta_{i\chi}(x),$ where $\{\chi_i\}$ forms a partition of the state space $\mathcal{X}$. Convergence is shown and tight error bounds are derived. In the same article, another algorithm in the style of Equation 1 is presented that employs an interpolative linear operator $\mathcal{E}$. Convergence and tight error bounds are derived.

Apparently, the only work known to us which does not make use of a model and which employs local learning is that of (Singh et al., 1995). The authors of these article propose a combination of Q-learning with what they call “soft-state aggregation”. The soft-clusters are represented by probability distributions $P(i|x) = \sum_{i=1}^{n} P(i|x) = 1,$ $P(i|x) > 0$. A fixed persistently exciting policy is used to generate a stationary state-action sequence $(X_t, A_t)$ over a finite state and action space. In each time-step a random cluster index $I_t$ is generated from $P(\cdot|X_t)$ and the Q-values are updated using the experience samples $< I_t, A_t, R_t, I_{t+1} >$. Almost sure convergence is guaranteed since $< I_t, A_t, R_t, I_{t+1} >$ has a stable statistics (Singh et al., 1995).

Although this algorithm is guaranteed to converge, it is quite inefficient as it updates only one $Q$-value in each time step even if the clusters “overlap” in an earlier work Q-learning was studied with “multi-state” updates (Ribeiro & Szepesvári, 1996; Szepesvári & Littman, 1999). In the algorithm considered multiple states are updated in each time step (hence the name of it), the original motivation being to share samples across “neighboring” Q-values to speed up the convergence process. Convergence results and bounds on the error introduced were given.

One main motivation of the present article is to extend this algorithm to large or even infinite state spaces. This is achieved by combining the ideas of (Tsitsiklis & Van Roy, 1996) with that of (Ribeiro & Szepesvári, 1996). The key idea is to employ interpolative function approximators. This way a familiar error recursion can be shown to hold for the error in the parameter space. Following the ideas of (Szepesvári & Littman, 1999), instead of showing the convergence of this combined algorithm directly we derive a generic convergence theorem of which the given algorithm will be shown to be a special case. Our results will show convergence of on-line learning algorithms even in infinite state-spaces. The actual form of the function approximator employed is left unspecified apart from the requirement that the corresponding operator must be a non-expansion.

The organization of the paper is as follows: In Section 2 the basic definitions are given and the notation is introduced. In Section 3 the form of the generic algorithm is proposed. The main convergence result along with its application to $Q$-learning and approximate value iteration is presented in Section 4. Conclusions are drawn and some open issues are discussed in Section 5.

2. Definitions

We assume that the reader is familiar with basic concepts of Markov decision processes (MDPs). In this section we introduce our notation, but do not provide any explanation of the concepts used.

2.1 MDPs

An MDP is a tuple $\mathcal{M} = (\mathcal{X}, A, P, R, \gamma)$, where $\mathcal{X}$ is a set of states, $A$ is a set of actions, $P$ is the transition probability density function for non-countable state spaces and the transition probability function for countable state spaces, $r$ is the reward function and $\gamma$ is the discount factor.

The optimal value function associated with the MDP $\mathcal{M}$ shall be denoted by $V^*$, the value operator by $T$, the optimal policy by $\pi^*$. The sup-norm shall be denoted by $\| \cdot \|$. The space of real-valued bounded functions over a set $\mathcal{X}$ will be denoted by $B(\mathcal{X})$.

2.2 Interpolative Function Approximation

A function approximator $F$ maps a parameter space $\Theta$ into the set of functions of interest, in our case $B(\mathcal{X})$, i.e., $F: \Theta \rightarrow B(\mathcal{X})$. Usually, $\Theta$ is a finite dimensional space, typically $\Theta = \mathbb{R}^k$ for some $k > 0$. For convenience, we shall denote the function $F(\theta)$ by $F_\theta$.

Besides choosing a function approximator, one also needs to choose a method to construct the parameter given some data. In this article we will consider interpolative methods, by which we mean the following: given the data $\mathcal{D} = (x_1, \ldots, x_n, (v_1, \ldots, v_n))$ with $x_i \neq x_j$, an interpolative method $R: \mathcal{X}^n \times \mathbb{R}^n \rightarrow \Theta$ chooses a parameter vector $\theta$ such that $F_\theta(x_i) = v_i$ for all $i = 1, 2, \ldots, n$. 


Fix the basis points \( x = (x_1, \ldots, x_n), x_i \neq x_j \). Given a function \( V \) of \( B(X) \) the interpolative representation of \( V \) as given by the pair \((F, R)\), where \( \theta = R(D_V) \) and \( D_V = (x_1, \ldots, x_n), (V(x_1), \ldots, V(x_n)) \). Let \( G : B(X) \rightarrow B(X) \) be the interpolative operator defined by \( GV = F(R(D_V)) \).

The mapping that “projects” functions in \( B(X) \) onto \( \mathbb{R}^n \) given the basis points \( x \) shall be denoted by \( \mathcal{P} \). \( \mathcal{P} : B(X) \rightarrow \mathbb{R}^n, \mathcal{P}(x_i) = V(x_i) \). Further, let us define \( \mathcal{E} : \mathbb{R}^n \rightarrow B(X) \) by \( \mathcal{E}V = GV \), where \( V \) is any function in \( B(X) \) satisfying \( V(x_i) = v_i, i = 1, 2, \ldots, n \). \( \mathcal{E} \) is well-defined, by the definitions of \( G \) and \( R \).

It is easy to see that these operators satisfy the equations

\[
\mathcal{E} \mathcal{P} V = GV, \quad (2)
\]
\[
\mathcal{P} \mathcal{E} = \text{id}_{\mathbb{R}^n}, \quad (3)
\]

where \( \text{id}_{\mathbb{R}^n} \) is the identity over \( \mathbb{R}^n \). Here the second equation states that \( G \) is interpolative. Note that \( \mathcal{P} \) is linear: \( \mathcal{P}(\lambda_1 U + \lambda_2 V) = \lambda_1 \mathcal{P}U + \lambda_2 \mathcal{P}V, U, V \in B(X), \lambda_1, \lambda_2 \in \mathbb{R} \). Further, \( |U| = |\mathcal{P}U| \), where \(|U|\) is defined by \(|U|(x) = |U(x)|\).

In the algorithms and the analysis below, for the sake of generality, we shall take \( G \) as our starting point together with its decomposition into the projection \( (\mathcal{P}) \) and “expansion” \( (\mathcal{E}) \) operators satisfying Equations 2 and 3. We shall require below that \( G \) be a non-expansion w.r.t the supremum norm. Given the decomposition above this is equivalent to requiring that \( \mathcal{E} \) is a non-expansion \( (\mathbb{R}^n \) with taken with the \( \ell_\infty \) norm:

**Proposition 2.1.** Let \( G, \mathcal{P}, \mathcal{E} \) satisfy 2 and 3, where \( \mathcal{P} \) is the projection operator for some set of basis points. Then \( G \) is a non-expansion if and only if \( \mathcal{E} \) is a non-expansion. Further,

\[
\|\mathcal{P}GU - \mathcal{P}GV\| = \|\mathcal{P}U - \mathcal{P}V\|, \quad (4)
\]
\[
\|\mathcal{P}U - \mathcal{P}V\| \leq \|U - V\|, \quad (5)
\]
\[
\|GU - GV\| \leq \|\mathcal{P}GU - \mathcal{P}GV\|. \quad (6)
\]

**Proof.** Assume that \( G \) is a non-expansion. Let \( u, v \in \mathbb{R}^n \) be arbitrary. Choose \( U, V \in B(X) \) such that \( \mathcal{P}U = u \) and \( \mathcal{P}V = v \) and \(|U - V| = |u - v|\). Then \( \|\mathcal{E}U - \mathcal{E}V\| = \|\mathcal{E} \mathcal{P}U - \mathcal{E} \mathcal{P}V\| = \|GU - GV\| \leq \|U - V\| = \|u - v\|\). This proves that \( \mathcal{E} \) is a non-expansion. Given, now, assume that \( \mathcal{E} \) is a non-expansion. Let \( U, V \in B(X) \) be arbitrary, \( \|GU - GV\| \leq \|\mathcal{P}U - \mathcal{P}V\| \leq \|U - V\| \).

The first two inequalities are trivial. The third one follows since \( \|GU - GV\| \leq \|\mathcal{P}U - \mathcal{P}V\| \) (since \( \mathcal{E} \) is a non-expansion) and \( \|\mathcal{P}U - \mathcal{P}V\| \) is equal to \( \|\mathcal{P}GU - \mathcal{P}GV\| \) by 4.

Many well known interpolation operators \( (\mathcal{E}) \) are non-expansions. Among them are the 0th order spline interpolation with \( \mathcal{E}u = \sum_{i=1}^n u_i \chi_{A_i} \), where \( X \) is the disjoint union of the sets \( A_1, \ldots, A_n \) and \( \chi_{A_i} \) is the characteristic function of \( A \) (corresponding to state aggregation), certain higher order spline-interpolations, and radial basis function (RBF) interpolation, which is a special case of kernel-based approximation. All these interpolation architectures assume a linear form: \( \mathcal{E}u = u^T \phi \) for a suitably chosen function \( \phi : X \rightarrow \mathbb{R}^n \). \( \phi \) is called the basis function and can either be explicitly or implicitly given. For example, if the sets \( A_i \) above are given explicitly then \( \phi = \{\chi_{A_1}, \ldots, \chi_{A_n}\} \). If \( A_i \) is given as the sets corresponding to the leaves of a decision tree then the basis function is implicit, – it is encoded into the decision tree algorithm. In the case of RBF networks \( \phi = U^{-T} \psi \), where \( \psi : X \rightarrow \mathbb{R}^n \) is a mapping such that \( \psi_i(x) = K(x, i, x) \) for some radially symmetrical kernel function \( K \), and \( U \) is defined by \( u_{ij} = \psi_i(x_j) = K(x, i, d(x_i, x_j)) \).

### 2.3 Asynchronous Stochastic Dynamic Programming

We will make use of the notion of the paper (Littman & Szepesvári, 1996) so as we can capture many asynchronous stochastic dynamic programming algorithms with a single algorithm and convergence proof. Let \( T_t : B \times B \rightarrow B \) be a sequence of stochastic operators satisfying certain regularity conditions. The algorithms considered in (Szepesvári & Littman, 1999) take the form

\[
V_{t+1} = T_t(V_t, V_t). \quad (7)
\]

In order to shed some light on the intuitions behind this notation consider (lookup table based) Q-learning. Take \( B = B(X \times A) \) and let

\[
[T_t(Q, Q')](x, a) = (1 - \alpha_t(x, a))Q(x, a) + \alpha_t(x, a) \left\{ R_t + \gamma \max_b Q'(X_{t+1}, b) \right\}, \quad (8)
\]

where \( \alpha_t(x, a) = 0 \) if \( (X_t, A_t) \neq (x, a) \). Here \((X_t, A_t)\) is the state-action pair visited at time step \( t \) and \( R_t \) is the immediate reward received. Note that the operator sequence \( T_t \) depends on the random trajectory \((X_0, A_0, R_0, X_1, \ldots)\) and is therefore random itself. The choice of \( T_t \) is motivated by the observation

---

2 Later we will relax this condition in the spirit of (Tsitsiklis & Van Roy, 1996).

3 Conditions under which the RBF interpolation operator of RBF networks exists and is a non-expansion are given in (Tsitsiklis & Van Roy, 1996).
that each component of $Q_{t+1} = T_t(Q_t, Q)$ is a standard one-dimensional stochastic approximation process and thus, under the usual conditions, $Q_t$ converges to $T_Q Q$ w.p.1, where $T_Q$ denotes the optimal value operator for $Q$-functions. Therefore algorithm 7 can be thought of as performing a “diagonal” approximation to $\lim_{t \to \infty} T_Q Q_0$.

Although, asynchronicity does not follow directly from the form of 7, using $Q$-learning as an example, one can see that the form of the algorithm allows the different components of the value function to be updated at different rates - hence the algorithm allows asynchronous updates.

In (Szepesvári & Littman, 1999) a general convergence result is presented for this algorithm. In the same article it was shown that this convergence result applies to many well known algorithms, including e.g. model-based reinforcement learning, asynchronous dynamic programming, $Q$-learning, $Q$-learning with multi-state updates. This last result will play an important role in this paper.

The convergence analysis of the algorithm builds on the comparison of the sequence $U_{t+1} = T_t(V_t, V^*)$ with $V_t$. In particular, a recursive error equation is derived for their difference.

3. The RLI Algorithm

The proposed RLI (“reinforcement learning with interpolation”) algorithm has the form

$$ V_{t+1} = T_t(V_t, V_t), $$

in the value function space, for a suitably chosen sequence of random operators and non-expansive interpolation operators $T_t$. This algorithm can be thought of as the extension of the algorithm proposed by Gordon who studied the iteration $V_{t+1} = GTV_t$. If one thinks of 7 as the generalization of value-interpolation then clearly 9 appears analogous to the algorithm studied by Gordon.

Given the decomposition $T_t = \epsilon_t P_t$, where $P_t$ is a projection operator as in the previous section, one readily derives the parameter space recursion

$$ \theta_{t+1} = P_t T_t(\epsilon_t \theta_t, \epsilon_t \theta_t), $$

where $V_t = \epsilon_t \theta_t$ or $\theta_t = P_t V_t$. We will be primarily concerned with the convergence of $V_t$ and the quality of approximation of the optimal value function by the limes. Note that if the limes exists and if $P_t = \mathcal{P}$ for some $\mathcal{P}$ and for all $t > 0$ then by the continuity of $\mathcal{P}$ and since $\theta_t = \mathcal{P} V_t$, $\theta_t$ will also converge to some limiting value.

4. Results

4.1 Convergence Analysis

For the simplicity of presentation let $G_t = G$, $P_t = \mathcal{P}$ and $\epsilon_t = \mathcal{E}$. Following the ideas developed in (Szepesvári & Littman, 1999) we compare $U_{t+1} = G T_t(U_t, V^*)$ with $V_{t+1} = G T_t(V_t, V_t)$. Here $V^*$ is the fixed point of $G T$. Let $\delta_t = | V_t - U_t |$ denote the error process.

First, note that by 6,

$$||\delta_{t+1}|| = ||G T_t(V_t, V_t) - G T_t(U_t, V^*)|| \leq ||P G T_t(V_t, V_t) - P G T_t(U_t, V^*)|| = ||P V_{t+1} - \mathcal{P} U_{t+1}|| = ||\mathcal{P} \delta_{t+1}||. \quad (11)$$

This inequality will play a key role in proving the convergence of $V_t$ as it shows that it is sufficient to prove that $\mathcal{P} \delta_t$ converges to zero w.p.1. This way, the problem is reduced to a finite dimensional problem.

Now, by 3 $P V_{t+1} = \mathcal{P} \mathcal{E} P T_t(V_t, V_t) = \mathcal{P} T_t(V_t, V_t)$. Similarly, $P \mathcal{P} U_{t+1} = \mathcal{P} T_t(U_t, V^*)$. Therefore

$$\mathcal{P} \delta_{t+1} = | P T_t(V_t, V_t) - \mathcal{P} T_t(U_t, V^*) |. \quad (12)$$

Now, recall that in (Szepesvári & Littman, 1999) the $T_t$ operators were shown to satisfy the following conditions in most of the applications:

$$| T_t(U_1, V) - T_t(U_2, V) | \leq G_t | U_1 - U_2 |, \quad (13)$$

and

$$| T_t(U_1, V) - T_t(U, V) | \leq F_t ( | V - V_t | + \lambda_t ), \quad (14)$$

where $U_1, U_2, V, V_2 \in B(\mathcal{X})$ are arbitrary, $\lambda_t \geq 0$ is a random sequence converging to zero w.p.1, $G_t, F_t$ are suitable chosen uniformly bounded non-negative random functions satisfying $0 \leq F_t, G_t \leq 1$, $F_t \leq \gamma (1 - G_t)$, and $\lim_{n \to \infty} \| T_{t_0} \| = 0$ w.p.1 for all $t_0 > 0$ and $0 < \gamma < 1$ is a discount factor (a non-random real number).

Note that in the applications $F_t$ typically corresponds to $\alpha_t \gamma$ and $G_t$ to $1 - \alpha_t$, where $\alpha_t$ is the “learning rate” at time step $t$ (cf. Equation 8). Proceeding formally, using 13 and 14 we get

$$\mathcal{P} \delta_{t+1} \leq | P T_t(V_t, V^*) - P T_t(U_t, V^*) | - | P T_t(V_t, V_t) - P T_t(U_t, V^*) | \leq P ( G_t | U_t - U_t | ) + P ( F_t ( | V_t - V_t | + | U_t - V^* | + \lambda_t ) ) = ( P ( G_t ) ( | U_t - U_t | ) + P ( F_t ) ( | \delta_t | + | U_t - V^* | + \lambda_t ).$$

4Here and in what follows multiplicaiton, absolute value, equality and inequality of functions should be understood pointwise.
4.3 Interpolatable Q-learning

Let \( G = \mathcal{EP} \) be any interpolatable, non-expansion defined over \( B \times A \).

Let \( X_0, A_0, R_0, X_1, A_1, R_1, \ldots \) be a random trajectory generated by a suitable stationary exploration policy \( (X_0 \text{ is the state visited at time } t, A_t \text{ is the action taken at time } t \text{ and } R_t \text{ is the reward received at time } t) \).

The straightforward combination of Q-learning and interpolatable function approximation generates the sequence \( Q_{t+1} = \mathcal{T}_t(Q_t, Q_t) \), where \( \mathcal{T}_t \) is the Q-learning update operator defined in Equation 8. This algorithm has the problem that it does not change \( Q_t \) unless \( (X_t, A_t) \in W \). Since this has a zero probability in uncountable state spaces the resulting algorithm will be rather uninteresting. We need to update \( Q \)-values at state-action pairs other than the currently visited ones!

In (Ribeiro & Szepesvári, 1996) and later in (Szepesvári & Littman, 1999), the process \( Q_{t+1} = \mathcal{T}_t(Q_t, Q_t) \) was studied where

\[
[\mathcal{T}_t(Q, Q')](z, a) = (1 - \alpha_t(z, a)s(z, a, X_t))Q(z, a) + \alpha_t(z, a)s(z, a, X_t) \left( R_t + \gamma \max_b Q'(X_{t+1}, b) \right),
\]

where \( \alpha_t \) is a suitably chosen sequence of random learning rates and \( s(z, a, x) > 0 \) is a spatial smoothing factor. Typically, \( s(z, a, x) \leq s(z, a, z) \) and \( s(z, a, x) \) decays to zero as \( \|z - z\| \to \infty \). We will require that \( s \) is measurable and uniformly bounded. In the updates we allow \( \alpha_t(z, a) > 0 \) for state-action pairs \((z, a) \neq (X_t, A_t)\). This way other state-action pairs than the currently visited one will be updated as well - just what we wanted to achieve.\(^8\)

In (Szepesvári & Littman, 1999) it was shown that in finite state spaces, under mild conditions \( Q_t \) defined above converges w.p.1. The motivation there, however, was to speed up Q-learning rather than making Q-learning work in infinite state spaces. The analysis required that every state action pair be visited infinitely often. Such a condition is never met in uncountable state spaces.

\(^6\)A special case is when the basis points \( (x_i, a_i) \) are chosen such that \( W = \{ (x_1, a_1), \ldots, (x_n, a_n) \} = \{ x_1, \ldots, x_n \} \times A \). This corresponds to approximation in the state space only. In what follows we shall assume this.

\(^7\)We will still need, however, \( \alpha_t(x, a) = 0 \) unless \( a = A_t \). This last condition could be removed by extending \( s \) to include “action smoothing” as well.

\(^8\)In (Singh et al., 1995) essentially the same was achieved by sampling which component to update. We believe that our method makes use of the samples more efficiently, but this still needs to be proven!
Here we show that this update rule when combined with interpolative function approximation yields a convergent algorithm. Let $Q_{t+1} = \mathcal{G} T(Q_t, Q_t)$, where $\mathcal{T}$ is defined by (16). In the parameter space the algorithm takes the form:

$$
\theta_{t+1,i} = (1 - \alpha_t(x_i, a_i)s(x_i, a_i, X_i))\theta_{t,i} + \alpha_t(x_i, a_i)s(x_i, a_i, X_i)
$$

$$
\left\{ R_t + \gamma \max_b Q_t(X_{t+1}, b) \right\}, \quad i = 1, 2, \ldots, n,
$$

(17)

where $Q_t = \mathcal{E}\theta_t$ and we have used $Q_t(x_i, a_i) = \theta_{t,i}$ which follows from the interpolative property of $\mathcal{G}$.

Now, let us consider $Q_{t+1} = \mathcal{T}(Q_t, Q_t)$, where $Q \in \mathcal{B}(X \times A)$ is fixed. In order to analyze this process we need to make some further assumptions on the sample path $\{ (X_t, A_t) \}$ and the policy $\pi$ that generates it. First, we require that $\pi(a|x) > 0$ for all $(x, a) \in X \times A$. Second, we require that $\{ X_t \}$ be stationary, to have a unique invariant measure, $\mu_X(\cdot)$, and let it be positive Harris (e.g., Tadić, 2001). It is known (see e.g., Tadić, 2001, Lemma 9) that under these conditions $(1/(t+1)) \sum_{s=0}^{t} f(X_s) \rightarrow \int f(x) d\mu_X(x)$, where $f$ is an arbitrary $L^1(\mu_X)$ function.

Now, by the positivity of $s$, $\int s(x_i, a_i, x)d\mu_X(x) > 0$ and thus there must exist some $\epsilon > 0$ such that $\mu_X(A) > 0$, where $A = \{ x : s(x_i, a_i, x) > \epsilon \}$. Let $n_t(x_i, a_i) = 1 + \sum_{s=0}^{t} \chi(s(x_i, a_i, X_t) > \epsilon)$. Clearly, $n_t(x_i, a_i) \rightarrow \infty$ w.p.1, since $(1/(t+1)) \sum_{s=0}^{t} \chi(s(x_i, a_i, X_t) > \epsilon) \rightarrow \mu_X(A)$. Therefore if $\alpha_t(x_i, a_i) = \chi(s(x_i, a_i, X_t) > \epsilon)/n_t(x_i, a_i)$ then $\sum_{s=0}^{t} \alpha_t(x_i, a_i)s(x_i, a_i, X_s) \geq \epsilon \sum_{k=0}^{\infty} n_t(x_i, a_i) / (k+1)$. For the same sequence, by the boundedness of $s$, $\sum_{s=0}^{t} \alpha_t(x_i, a_i)s(x_i, a_i, X_s) \rightarrow \infty$ w.p.1.

Now, the analysis presented for multi-state Q-learning in (Szepesvári & Littman, 1999) applies to $\theta_t = \mathcal{P} \bar{Q}_t$ and shows that it converges to $(\mathcal{P}\mathcal{H})(Q)$, where $\mathcal{H} : \mathcal{B}(X \times A) \rightarrow \mathcal{B}(X \times A)$ is given by

$$
\mathcal{H}(Q)(z, a) = \int \int \tilde{s}(z, a, x)(R(x, a) +
\gamma \max_b Q(y, b)p(y|x, a)dy)d\mu_X(x),
$$

(18)

where $\tilde{s}(z, a, x) = s(z, a, x)/(\int s(z, a, x)d\mu_X(x))$. By standard arguments $\mathcal{H}$ can be shown to be a contraction operator with contraction factor $\gamma$.

Now let $\mathcal{T} : \mathcal{B}(X \times A) \rightarrow \mathcal{B}(X \times A)$ be defined by $\mathcal{T} = \mathcal{G}\mathcal{H}$ and let $\mathcal{Q}^*$ be the fixed point of it. We claim that $Q_{t+1} = \mathcal{G} \mathcal{T}(Q_t, Q_t)$ converges to $\mathcal{Q}^*$ w.p.1. We shall make use of Theorem 4.1. Let $F_t(z, a) = \gamma \alpha_t(z, a)s(z, a, X_t)$ and $G_t(z, a) = (1 - \alpha_t(z, a)s(z, a, X_t))$. $\lim_{n \rightarrow \infty} \| (\mathcal{P}^n \mathcal{G} \mathcal{T}) \| = 0$ follows iff $\sum_{i=0}^{\infty} \alpha_t(z, a)i = \infty$ w.p.1., $i = 1, 2, \ldots, n$. This proves the following theorem:

**Theorem 4.3.** Fix a stochastic exploration policy $\pi(a|x)$ such that the sequence of the generated states $X_t$ has a unique invariant measure $\mu_X$, $\{ X_t \}$ is stationary and positive Harris. Assume further that $\pi(a|x) > 0$ for all $a \in A$ and $x \in X$. Let $\mathcal{E}$ be a non-expansion, let $\alpha_t(z, a)$ be defined as in the previous paragraph, where $s(z, a, x) > 0$ is a design parameter. Let $s$ be measurable and bounded. Then $\theta_{t+1,i}$ converges w.p.1 to $\theta^*$ such that $\mathcal{Q}^* = \mathcal{E}\theta^*$ is the fixed point of $\mathcal{G}\mathcal{H}$, where $\mathcal{H}$ is defined by (18).

Since TD(0) is a special case of Q-learning (where the cardinality of $A$ is one), one obtains a convergence result for TD(0) as a corollary of the above theorem. This result complements those that were obtained by Van Roy and Tsitsiklis (Tsitsiklis & Van Roy, 1997) and more recently by Tadić for gradient-based TD($\lambda$) (Tadić, 2001).

### 4.4 Interpolative Approximate Asynchronous Value Iteration

Let us consider again the iteration $V_{t+1} = \mathcal{G} \mathcal{V}_t$ studied by (Gordon, 1995), or the equivalent form (Equation 1) in the parameter space. Convergence can be proved as the composite operator $\mathcal{G}\mathcal{T}$ is a contraction with the same index as that of $\mathcal{T}$.

However, this algorithm is not practical if the number of states accessible from $x_i$ (the $i$th base-point) is very large or infinite. In (Tsitsiklis & Van Roy, 1996) the authors consider the same algorithm was considered for interpolative representations. As another application of Theorem 4.1 the convergence of an extension of the second algorithm consider in (Tsitsiklis & Van Roy, 1996) is derived here.

There are a number of ways to approximate $\mathcal{P}\mathcal{T}\mathcal{E}\theta_t$. An obvious approximation is to replace $T$ by a Monte-Carlo approximation of it. If $T$ is “randomized” in this way then one must change $\theta_t$ gravely to filter out the effects of noise:

$$
\theta_{t+1} = (1 - \alpha_t)\theta_t + \alpha_t\mathcal{P}\mathcal{T}\mathcal{E}\theta_t.
$$

One possibility is to sample $K$ transitions $\{(X_{i,k}, R_{i,k})\}_{k=1,\ldots,K}$ from each basis point $x_i$ and action $a$ and define $(\mathcal{P}\mathcal{T}\mathcal{V})_i = (1/N)\sum_{k=1}^{N} \{R_{i,k} + \gamma V(X_{i,k})\})$. As before, the algorithm can be put in the form $V_{t+1} = \mathcal{T}(V_t, V_t)$ and analyzed using the tools developed above. Under

---

9 At this level of generality one does not need the interpolative requirement.
the usual conditions on $\alpha_t$, the samples and $L$, one obtains convergence to the fixed point of $GT$ w.p.1.

If one does not have a sampling device that can generate samples from any point of interest then one may use the idea of spatial smoothing and rely on a single sample path generated using a persistently exciting policy. Again, the resulting algorithm can be proven to converge under conditions similar to that of in Theorem 4.3.

Many other variants can be proposed. Due to the lack of space we cannot discuss these here.

4.5 Adaptive Representations

Deciding for a good representation for a given MDP requires good insight of the problem to be solved. Therefore it is important to consider methods that refine the representation on the basis of observed data. The iteration $V_{t+1} = G_t T_t(V_t, V_t)$ models such changing representations provided that we allow the operator $G_t$ to become random (since it will depend on random data). Examples that can be cast into the present framework include refinements of triangulations as in (Munos, 1997), decision tree growing (Wang & Dietterich, 1999) and growing kernel-based methods (Smart & Kaelbling, 2000).

If there exist a deterministic bound on the number of changes of the representation (e.g. adaptive discretization is stopped when a given maximal spatial resolution is achieved or when a maximum number of allowable changes is achieved) then $V_t$ will converge to $V^*$, the fixed point of $G_\infty T$, where $G_\infty = \lim_{t \to \infty} G_t$. Note that in general $G_\infty$ will be random, but the main theorem continues to hold.

4.6 Relaxing the interpolation requirement

The requirement that $G$ should be a non-expansion can be relaxed if one observes that all what matters is that $G T$ should be a contraction. For this it is sufficient to require that $G$ is Lipschitz with a constant $0 < \gamma' \leq 1 / \gamma$ (or $\| E V - E U \| \leq \gamma' \| U - V \|$), where $U, V \in B(X)$ are arbitrary). This was first observed in (Tsitsiklis & Van Roy, 1996). The results derived here can also be shown to hold under this relaxed condition.

5. Conclusion

We have introduced a class of general interpolative reinforcement learning algorithms and studied their convergence. In particular, an extension of Q-learning was studied in uncountable state spaces. Almost sure convergence was derived and bounds were given to the loss resulting from using the given architecture.

One attractive feature of Q-learning is that it allows the exploration policy to be chosen at will as long as it remains “infinitely exploring” (see (Singh et al., 1997)). Unfortunately, the result of the previous section requires that the exploration policy be fixed during learning. A naive approach to overcome this problem is to multiply $\alpha_t(z,a)s(z,a,X_t)$ in the update equation with a correction factor $C_t$, where $C_t$ is obtained recursively: $C_{t+1} = C_t(\pi(A_{t+1}|X_{t+1})/\pi_t(A_{t+1}|X_{t+1})$. Here $\pi$ is a fixed exploration policy and $\pi_t$ is the policy that is actually followed during learning. Trivial computation shows that for any real-valued, bounded measurable function $f$, $E[f(X_t)C_t] = \int f(x)d\mu_X(x)$, i.e., $C_t$ successfully compensates for the changes in the policy. Unfortunately, there is a catch: $C_t$ in general cannot be kept bounded and uniformly positive and thus the previous convergence results will not continue to hold. Therefore decoupling exploration and learning remains a very interesting open problem.

If there is no way to decouple learning from the exploration policy then one may hope to use to use approximate policy iteration together with the considered methods. Then the rate of convergence of the method becomes very important.

Finally, in connection to adaptive interpolation methods, some interesting problems are as follows: (i) Derive conditions under which (specific) algorithms let $G_t$ converge to some deterministic $G_\infty$. (ii) Extend the analysis to the case of indefinitely growing representations, derive consistency results.

References


