Interpolation-based Q-learning

Csaba Szepesvári SZCSABA@SZTAKI.HU
Computer and Automation Research Institute of the Hungarian Academy of Sciences,
1111 Budapest XI, Kende u. 13-17, Hungary.

William D. Smart WDS@CSE.WUSTL.EDU
Department of Computer Science and Engineering,
Washington University in St. Louis, One Brookings Drive, St. Louis, MO 63130, United States.

Abstract

We consider a variant of Q-learning in continuous state spaces under the total expected discounted cost criterion combined with local function approximation methods. Provided that the function approximator satisfies certain interpolation properties, the resulting algorithm is shown to converge with probability one. The limit function is shown to satisfy a fixed point equation of the Bellman type, where the fixed point operator depends on the stationary distribution of the exploration policy and the function approximation method. The basic algorithm is extended in several ways. In particular, a variant of the algorithm is obtained that is shown to converge in probability to the optimal Q function. Preliminary computer simulations are presented that confirm the validity of the approach.

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. Reinforcement learning (RL) is sometimes defined as large-scale approximate dynamic programming combined with learning techniques. Indeed, RL applied to continuous state spaces has been a long standing challenging problem.

In this paper we propose and study methods that allow Q-learning to work in continuous state spaces, under the total expected discounted cost criterion. In its very basic form our method updates the parameters of some function approximator, where the update equations take a slightly modified form of the Q-learning equations. In order to be more specific, let (X_t, A_t, R_t) be a controlled Markov process, X_t being the state visited, A_t being the action chosen ($A_t \in A$) and $R_t \in \mathbb{R}$ being the reward observed at time step t. Further, let $0 < \gamma < 1$ be a discount factor. We assume that $X_t \in \mathcal{X}$, where the state space $\mathcal{X}$ is a compact subset of a Euclidean space $\mathbb{R}^d$ ($d \geq 1$). We also assume that the action space A is finite. Denoting by θ_t the parameters of the function approximator at time step t ($\theta_t \in \mathbb{R}^n$, where n is the number of parameters), our basic algorithm takes the following form:

$$\Delta\theta_{ti} = \alpha_{ti} s_{ti} \left(R_t + \gamma \max_b F_{\theta_t}(X_{t+1}, b) - \theta_{ti} \right). \quad (1)$$

Here α_{ti} is the learning rate associated with component i at time t, the factor s_{ti} depends on X_t and i and determines how much influance sample X_t has on updating component i, F denotes a function approximator with $F_\theta : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$. At the price of a slight abuse of notation we also write $F(\theta)$ instead of F_θ when we want to emphasize the dependency of F on the parameter vector θ. In this sense, F also represents a mapping that maps the parameter space $\mathbb{R}^n$ to real-valued functions defined over $\mathcal{X} \times \mathcal{A}$. Throughout this article we assume that F is a non-expansion that satisfies a system of interpolation equations with respect to some fixed set of basis points S of cardinality n. Specifically, we assume that if $S = \{(x_1, a_1), \ldots, (x_n, a_n)\}$ then $F_\theta(x_i, a_i) = \theta_i$ holds, for all $i = 1, \ldots, n$. This holds e.g. if $S = \{x_1, \ldots, x_p\} \times A$ and if $F_\theta(\cdot, a)$ is a barycentric interpolator defined over a triangulation induced by the set of basis points $\{x_1, \ldots, x_p\}$ for each $a \in A$. We shall assume that the values s_{ti} are defined

Appearing in *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004.

by the equations

$$s_{ti} = s(x_i, a_i, X_t), \quad i = 1, \dots, n, \tag{2}$$

where $s : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to \mathbb{R}$ is a bounded measurable spatial smoothing function. Typically $s(x, a, z)$ will be smooth and decays to zero as $\|x - z\|$ tends to infinity. One possible choice is to use a Gaussian function. Note that Equation (1) allows the update of multiple components of θ_t unlike in standard Q-learning.

Our main results will be the following: Initially, assume that a stochastic stationary policy is fixed that is used to sample the states (X_t) and actions (A_t). Further, assume that this policy is such that (X_t) is a sufficiently regular Markov process. Then θ_t converges with probability one to some parameter vector θ^* such that F_{θ^*} satisfies a fixed point equation of the Bellman type. This result is then extended to methods that add new basis points in an adaptive manner. Finally, we briefly outline a multi-stage method that is claimed to yield estimates that converge to the optimal value function. The same multi-stage method allows one to relax the condition of having to use a fixed sampling policy during the course of learning. We also present some experimental results where we provide a preliminary comparision of a particular instantiation of the proposed algorithm and some related algorithms from the literature.

2. Definitions

We assume that the reader is familiar with the concepts of reinforcement learning. Here we introduce only the necessary notation and some well known facts. The sup-norm shall be denoted by $\|\cdot\|$. The space of real-valued bounded functions over a set X will be denoted by $B(X)$. Except where otherwise noted multiplication, absolute value, equality and inequality of functions are defined componentwise. An operator $T : X \to Y$, where X and Y are metric spaces is called a γ-contraction if $\|Tf - Tg\| \leq \gamma \|f - g\|$ holds for all $f, g \in X$. 1-contracting operators are called non-expansive. The equation $Tf = f$ is called a fixed point equation. Any f satisfying $Tf = f$ is called the fixed point of T. When $T : X \to X$ is a γ-contraction with $0 < \gamma < 1$ and X is a complete metric space then T has a unique fixed point.

By an MDP we mean a 5-tuple $\mathcal{M} = (\mathcal{X}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{X}$ is a set of states, $\mathcal{A}$ is a set of actions, P is a transition probability law, r is a reward function and γ is a discount factor.

3. Interpolative Non-expansions

A function approximator allows one to use finite dimensional spaces to represent functions with continuous domains. In this sense a function approximator maps the parameter space $\mathbb{R}^n$ to the space of functions defined over some domain X. Note that at this level of generality we are not interested in how a given parameter $\theta \in \mathbb{R}^n$ is obtained, but we are only interested in the properties of the mapping that assigns functions to the parameters. The idea of using function approximators to solve fixed point equations, first proposed in the RL literature by Gordon (1995) is as follows: Assume that we are interested in the fixed point of T that maps the space of bounded functions into itself. Let $\mathcal{P}$ be a mapping that maps functions from $B(X)$ into $\mathbb{R}^n$. Conversely, let F map points of $\mathbb{R}^n$ into functions of $B(X)$. For obvious reasons, we shall call such mappings "function approximators". Then define the algorithm

$$\theta_{t+1} = \mathcal{P}TF\theta_t, \quad \theta_0 \in \mathbb{R}^n, t = 0, 1, 2, \dots. \tag{3}$$

Note that if for any $V \in B(X)$, $\mathcal{P}TV$ can be computed with a finite amount of work then this algorithm can be implemented using finite resources. Now consider the iteration

$$V_{t+1} = TF\mathcal{P}V_t \tag{4}$$

in the space $B(X)$ and define $\theta_t = \mathcal{P}V_t$. Then we see that θ_t satisfies (3) and iteration (3) can be thought of as a finite dimensional approximation of value iteration. It is not too hard to show that if T is a contraction and $F\mathcal{P}$ is a non-expansion then θ_t converges to some limit value θ^* such that $F\theta_t$ converges to $F\theta^* = V^*$, where V^* is the (unique) fixed point of the composite operator $F\mathcal{P}T$ (Gordon, 1995).

In order to extend the above idea to the learning scenario in a rigorous manner we will need some more assumptions on F. In particular, the notion of interpolative function approximators will be central to our analysis. In order to introduce this concept, we first need to be more specific about the operator $\mathcal{P}$. For this, fix a finite set of basis points $S = \{x_1, \dots, x_n\} \subset X$ and define $\mathcal{P} : B(X) \to \mathbb{R}^n$ by

$$(\mathcal{P}V)_i = V(x_i), \quad i = 1, 2, \dots, n. \tag{5}$$

Then $\mathcal{P}$ is called the composite point evaluation operator w.r.t. S. Using this notion we can now define the notion of interpolative function approximators.

Definition 3.1. *Let $F : \mathbb{R}^n \to B(X)$ be a mapping that maps parameters to functions. Then F is called* interpolative *w.r.t. the set of basis points S if it holds that for all functions $V \in B(X)$ $\mathcal{P}F\mathcal{P}V = \mathcal{P}V$, or*

$$\mathcal{P}F\mathcal{P} = \mathcal{P}. \tag{6}$$

That is, F is interpolative w.r.t. to the set of basis points S when for all parameters θ the function $F(\theta)$ takes on the value θ_i when evaluated at the point x_i, $i = 1, \ldots, n$:

$$F_\theta(x_i) = \theta_i, \quad i = 1, 2, \ldots, n.$$

Actually, this condition is equivalent to the above definition. It should be obvious by now why F is called interpolative.

Since we need to go in both directions between the spaces $\mathbb{R}^n$ and $B(X)$ in an alternating manner it will be useful to define the composite mapping $\mathcal{G} : B(X) \to B(X)$, $\mathcal{G} = F\mathcal{P}$. Note that using $\mathcal{G}$ Equation (6) takes the form $\mathcal{P}\mathcal{G} = \mathcal{P}$. Further $\mathcal{G}$ satisfies property (I): $\mathcal{G}V = \mathcal{G}V'$ whenever V, V' are such that $\mathcal{P}V = \mathcal{P}V'$. Equivalently, one may start with a mapping $\mathcal{G} : B(X) \to B(X)$ that satisfies $\mathcal{P}\mathcal{G} = \mathcal{P}$ and property (I) and define $F : \mathbb{R}^n \to B(X)$ such that $F\mathcal{P}V = \mathcal{G}V$. Then F is well-defined by (I) and since $\mathcal{P}$ is surjective. In such a case $\mathcal{G}$ is also called an interpolative mapping. The following proposition summarizes some of the basic properties of interpolative non-expansions:

Proposition 3.2 (Basic properties of interpolative non-expansion mapping). *Let $\mathcal{P}$ be the composite point evaluation operator over $B(X)$ w.r.t. some basis point set S. Let $\mathcal{G} : B(X) \to B(X)$ and $F : \mathbb{R}^n \to B(X)$ be mappings such that the equations*

$$F\mathcal{P} = \mathcal{G}, \tag{7}$$
$$\mathcal{P}F = \mathrm{id}_{\mathbb{R}^n}, \tag{8}$$

are satisfied, where $\mathrm{id}_{\mathbb{R}^n}$ is the identity operator over $\mathbb{R}^n$. Then (i) $\mathcal{G}$ is a non-expansion if and only if F is a non-expansion. Further, if F is a non-expansion then (ii):

$$\|\mathcal{P}\mathcal{G}U - \mathcal{P}\mathcal{G}V\| = \|\mathcal{P}U - \mathcal{P}V\|, \tag{9}$$
$$\|\mathcal{P}U - \mathcal{P}V\| \le \|U - V\|, \tag{10}$$
$$\|\mathcal{G}U - \mathcal{G}V\| = \|\mathcal{P}\mathcal{G}U - \mathcal{P}\mathcal{G}V\|. \tag{11}$$

Proof. Assume that $\mathcal{G}$ is a non-expansion. Now, 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 $\|Fu - Fv\| = \|F\mathcal{P}U - F\mathcal{P}V\| = \|\mathcal{G}U - \mathcal{G}V\| \le \|U - V\| = \|u - v\|$. This proves that F is a non-expansion. Now, assume that F is a non-expansion. Let $U, V \in B(X)$ be arbitrary. Then $\|\mathcal{G}U - \mathcal{G}V\| \le \|\mathcal{P}U - \mathcal{P}V\| \le \|U - V\|$. This finishes the proof of part (i).

Now, assume that F is a non-expansion. Equations (9) and (10) are trivial. Equation (11) follows since $\|\mathcal{G}U - \mathcal{G}V\| \le \|\mathcal{P}U - \mathcal{P}V\|$ (since F is a non-exponasion) and $\|\mathcal{P}U - \mathcal{P}V\|$ is equal to $\|\mathcal{P}\mathcal{G}U - \mathcal{P}\mathcal{G}V\|$ by (9). Hence $\|\mathcal{G}U - \mathcal{G}V\| \le \|\mathcal{P}\mathcal{G}U - \mathcal{P}\mathcal{G}V\| \le \|\mathcal{G}U - \mathcal{G}V\|$, where the last inequality follows by (10). □

4. Main Results

We start with an extension of a result due to Szepesvári and Littman (1999). The extension concerns the convergence of a sequence of random functions V_t that satisfy the iteration

$$V_{t+1} = \mathcal{G}\mathcal{T}_t(V_t, V_t), \quad V_0 \in B(X), t = 0, 1, 2, \ldots. \tag{12}$$

Here $\mathcal{T}_t$ is a sequence of random operators mapping $B(X) \times B(X)$ into $B(X)$, and $\mathcal{G} : B(X) \to B(X)$ is assumed to be a non-expansive interpolative mapping. Intuitively, (12) can be thought of as a randomized version of approximate value function iteration (4) (i.e. the operators $\mathcal{T}_t$ are randomized versions of T). Although at a first glance, it looks odd that $\mathcal{T}_t$ is a two argument mapping and that in iteration (12) both arguments are the same, this special two argument form allows one to reduce the convergence properties of asynchronous algorithms to synchronous once, as we shall see it soon.

Given the decomposition $\mathcal{G} = F\mathcal{P}$, where $\mathcal{P}$ is the composite point evaluation mapping, one readily derives the parameter space recursion

$$\theta_{t+1} = \mathcal{P}\mathcal{T}_t(F\theta_t, F\theta_t), \tag{13}$$

from Equation (12), where $V_t = F\theta_t$, $t = 1, 2, \ldots$ and $\theta_t = \mathcal{P}V_t$, $t = 0, 1, 2, \ldots$. This can be proven e.g. by defining $\theta_t = \mathcal{P}V_t$. Then by the interpolative property of F, we have that $\theta_{t+1} = \mathcal{P}V_{t+1} = \mathcal{P}F\mathcal{P}\mathcal{T}_t(V_t, V_t) = \mathcal{P}\mathcal{T}_t(V_t, V_t)$. Further, $V_{t+1} = \mathcal{G}\mathcal{T}_t(V_t, V_t) = F\mathcal{P}\mathcal{T}_t(V_t, V_t) = F\theta_{t+1}$. Since this holds for all $t = 0, 1, 2, \ldots$, we also have that $V_t = F\theta_t$ holds for $t = 1, 2, \ldots$. Putting the pieces together we arrive at (13). Note that (13) can be a practical algorithm (that can be carried out with finite resources), which is not that obvious and indeed does not hold in general for (12).

We will be primarily concerned with the convergence of V_t (equivalently, with the convergence of θ_t) and the quality of approximation of the optimal value function by the limit value. Note that if the limit exists then by the continuity of $\mathcal{P}$ and since $\theta_t = \mathcal{P}V_t$, θ_t will also converge to some limiting value.

Here is our first result:

Theorem 4.1. *Let $T : B(X) \to B(X)$ be a contraction with contraction coefficient $0 < \gamma < 1$ and let*

$\mathcal{G}$ be a non-expansive interpolative mapping w.r.t. to the composite point evaluation mapping $\mathcal{P}$. Let $\hat{V}^*$ be the fixed point of $\mathcal{G}T$. Then, iteration (12) *converges to $\hat{V}^*$ independently of V_0 provided that the following conditions hold: (i) The sequence $U_{t+1} = \mathcal{G}\mathcal{T}_t(U_t, \hat{V}^*)$ converges to $\hat{V}^*$ with probability one (w.p.1), independently of U_0. (ii) There exists a sequence of random function $G_t, F_t \in B(X)$ that satisfy $0 \le \mathcal{P}F_t, \mathcal{P}G_t \le 1$, $\mathcal{P}F_t \le \gamma(1 - \mathcal{P}G_t)$, and $P(\lim_{n\to\infty} \|\Pi_{t=t_0}^n \mathcal{P}G_t\| = 0) = 1$, where $t_0 \ge 0$ is an arbitrary natural number, and where $\mathcal{T}_t, F_t, G_t$ satisfy the following inequalities componentwise:*

$$|\mathcal{T}_t(U_1, V) - \mathcal{T}_t(U_2, V)| \le G_t|U_1 - U_2|, \quad (14)$$
$$|\mathcal{T}_t(U, V_1) - \mathcal{T}_t(U, V_2)| \le F_t\left(\|V_1 - V_2\| + \lambda_t\right). \quad (15)$$

Here $\lambda_t \ge 0$ is a random process that converges to zero w.p.1 and $U, U_1, U_2, V, V_1, V_2 \in B(X)$ are arbitrary.

Proof. The proof follows the steps of the main result of (Szepesvári & Littman, 1999): we compare $U_{t+1} = \mathcal{G}\mathcal{T}_t(U_t, \hat{V}^*)$ with $V_{t+1} = \mathcal{G}\mathcal{T}_t(V_t, V_t)$. Let $\delta_t = |V_t - U_t|$ denote the error process. First, note that by (11),

$$\begin{aligned}\|\delta_{t+1}\| &= \|\mathcal{G}T_t(V_t, V_t) - \mathcal{G}T_t(U_t, \hat{V}^*)\| \\ &= \|\mathcal{P}\mathcal{G}T_t(V_t, V_t) - \mathcal{P}\mathcal{G}T_t(U_t, \hat{V}^*)\| \\ &= \|\mathcal{P}V_{t+1} - \mathcal{P}U_{t+1}\| = \|\mathcal{P}\delta_{t+1}\|. \quad (16)\end{aligned}$$

This equality plays 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 (8) $\mathcal{P}V_{t+1} = \mathcal{P}F\mathcal{P}\mathcal{T}_t(V_t, V_t) = \mathcal{P}\mathcal{T}_t(V_t, V_t)$. Similarly, $\mathcal{P}U_{t+1} = \mathcal{P}\mathcal{T}_t(U_t, \hat{V}^*)$. Therefore

$$\mathcal{P}\delta_{t+1} = |\mathcal{P}\mathcal{T}_t(V_t, V_t) - \mathcal{P}\mathcal{T}_t(U_t, \hat{V}^*)|. \quad (17)$$

Proceeding formally, using (14) and (15) we get

$$\begin{aligned}\mathcal{P}\delta_{t+1} &\le |\mathcal{P}\mathcal{T}_t(V_t, \hat{V}^*) - \mathcal{P}\mathcal{T}_t(U_t, \hat{V}^*)| + \\ &\quad |\mathcal{P}\mathcal{T}_t(V_t, V_t) - \mathcal{P}\mathcal{T}_t(V_t, \hat{V}^*)| \\ &\le \mathcal{P}(G_t|V_t - U_t|) + \\ &\quad \mathcal{P}(F_t(\|V_t - U_t\| + \|U_t - \hat{V}^*\| + \lambda_t)) \\ &= (\mathcal{P}G_t)\left(\mathcal{P}|V_t - U_t|\right) + \\ &\quad (\mathcal{P}F_t)\left(\|\delta_t\| + \|U_t - \hat{V}^*\| + \lambda_t\right).\end{aligned}$$

Now, by (16), $\|\delta_t\| = \|\mathcal{P}\delta_t\|$ and therefore

$$\mathcal{P}\delta_{t+1} \le (\mathcal{P}G_t)\left(\mathcal{P}\delta_t\right) + (\mathcal{P}F_t)\left(\|\mathcal{P}\delta_t\| + \|U_t - \hat{V}^*\| + \lambda_t\right).$$

Notice that we have reduced the infinite dimensional error recursion to a finite dimensional one. Now, if $\mathcal{P}G_t$, $\mathcal{P}F_t$ satisfy $0 \le \mathcal{P}F_t, \mathcal{P}G_t \le 1$, $\mathcal{P}F_t \le \gamma(1 - \mathcal{P}G_t)$, and $\lim_{n\to\infty} \|\Pi_{t=t_0}^n \mathcal{P}G_t\| = 0$ w.p.1, for all $t_0 > 0$ and since $\|U_t - \hat{V}^*\|$ converges to zero, by Lemma 4 of (Szepesvári & Littman, 1999) $\mathcal{P}\delta_t$ converges to zero w.p.1. Hence, by (16) we also have $\|\delta_t\| \to 0$ w.p.1. Since U_t is known to converge to $\hat{V}^*$, it follows that $V_t \to \hat{V}^*$ w.p.1., as $t \to \infty$ holds, as well. □

We note in passing that although in this paper we are only concerned with the convergence of Q-learning, the above theorem is rather general and can be used to deduce the convergence of other reinforcement learning algorithms in continuous spaces when they are combined with interpolative function approximators.

Similarly to the above analysis, the convergence of U_t can be studied by looking at $\mathcal{P}U_t$. Namely, if $\mathcal{P}U_t$ converges to $\hat{\theta}^*$ then by the continuity of F, U_t must also converge w.p.1. Further, if $\hat{V}^*$ is the fixed point of the operator $\mathcal{G}T$ then $\|U_{t+1} - \hat{V}^*\| = \|\mathcal{G}\mathcal{T}_t(U_t, \hat{V}^*) - \mathcal{G}\hat{V}^*\| = \|\mathcal{P}U_{t+1} - \mathcal{P}\hat{V}^*\|$ by (11). This shows that if $\mathcal{P}U_t$ converges to $\mathcal{P}\hat{V}^*$ then U_t converges to $\hat{V}^*$ (and vice versa). This observation will be exploited in our next result where we study the convergence of the basic algorithm (1).

For the proof of our main result, we will need the following assumptions. Let $S = \{(x_1, a_1), \ldots, (x_n, a_n)\}$ be the basis point set.

Assumption A1 $(\mathcal{X}, \mathcal{A}, P, r, \gamma)$ is a discounted MDP, where $\mathcal{A}$ is finite, $\mathcal{X}$ is a compact subset of a Polish space, and (in order to play it safe) we assume that r is a continuous function of its arguments.

Assumption A2 $A_t \sim \pi(a = \cdot|X_t)$, where π is a fixed stochastic stationary policy satisfying $\pi(a|x) > 0$ over $\mathcal{X} \times \mathcal{A}$; $X_{t+1} \sim dP(\cdot|X_t, A_t)$, $X_0 \sim \pi_0$ for some probability measure π_0; (X_t) is a positive Harris, aperiodic chain (Meyn et al., 1996),[1][2] R_t is a stochastic process with uniformly bounded variance given $H_t = (X_t, A_t, R_{t-1}, X_{t-1}, A_{t-1}, \ldots, R_0, X_0, A_0)$ and $E[R_t|H_t] = r(X_t, A_t)$.

Assumption A3 For all $t = 0, 1, 2, \ldots$ and $i = 1, 2, \ldots, n$, $s_{ti} = s(x_i, a_i, X_t)$, where $s \ge 0$ is a bounded measurable function, such that $\int s(x_i, a_i, z) d\mu_X(z) > 0$ where μ_X is the unique invariant measure underlying the Markov chain (X_t).[3]

[1] This property is the analogue of positive recurrence defined for finite state space Markov chains.

[2] Note that we do not assume that the chain is stationary.

[3] Note that μ_X is the analogue of the stationary distribution for finite Markov chains. The existense and uniqueness of μ_X follow since (X_t) is positive Harris (Meyn et al.,

Assumption A4 For all $t = 0, 1, 2, \ldots$ and $i = 1, 2, \ldots, n$, $\alpha_{ti} = \chi(s(x_i, a_i, X_t) > \epsilon)/n_t(x_i, a_i)$, where $n_t(x_i, a_i) = 1 + \sum_{s=0}^{t} \chi(s(x_i, a_i, X_s) > \epsilon)$. Here $\chi(L) = 1$ iff the expression L holds true and $\chi(L) = 0$ otherwise. The constant $\epsilon > 0$ is selected such that $\mu_X(A_i) > 0$ holds for all $i = 1, 2, \ldots, n$, where $A_i = \{ z \mid s(x_i, a_i, z) > \epsilon \}$.

Note that Assumption A4 is the analogue of the condition widely used for finite models that every state is visited infinitely often. Let $s_\epsilon(x, a, y) = \chi(s(x, a, y) > \epsilon) s(x, a, y)$ denote the "ϵ-cut" of s. The following theorem holds true:

Theorem 4.2. *Consider the sequence θ_t, generated using (1) and assume that Assumptions A1–A4 hold. Further, assume that F is an interpolative nonexpansion w.r.t. the set of basis points S and let $\mathcal{G} = F\mathcal{P}$. Then θ_t converges to θ^* w.p.1 such that $\hat{Q}^* = F\theta^*$ is the fixed point of the operator $\mathcal{GH}$, where $\mathcal{H} : B(\mathcal{X} \times \mathcal{A}) \to B(\mathcal{X} \times \mathcal{A})$ is given by*

$$\mathcal{H}(Q)(z, a) = \int \int \hat{s}(z, a, x) \{ r(x, a) + \gamma \max_b Q(y, b) \} dP(y|x, a) d\mu_X(x). \tag{18}$$

Here $\hat{s}(z, a, x) = s_\epsilon(z, a, x) / (\int s_\epsilon(z, a, x) d\mu_X(x))$.

Proof. Let us define the process $Q_{t+1} = \mathcal{GT}_t(Q_t, Q_t)$, where

$$[\mathcal{T}_t(Q, Q')](z, a) = (1 - \alpha_t(z, a)) s(z, a, X_t) Q(z, a) + (1 - \alpha_t(z, a)) s(z, a, X_t) \left\{ R_t + \gamma \max_b Q'(X_{t+1}, b) \right\}. \tag{19}$$

Note that here α_t is defined over the whole space $\mathcal{X} \times \mathcal{A}$. However, by the special form of the iteration defining Q_t, one only needs to define the values of $\alpha_t(x_i, a_i)$. This follows from the identity

$$(\mathcal{PT}_t(Q, Q'))_i = (1 - \alpha_{ti}) s_{ti} Q(x_i, a_i) + (1 - \alpha_{ti}) s_{ti} \left\{ R_t + \gamma \max_b Q'(X_{t+1}, b) \right\} \tag{20}$$

and since by our previous observations $\theta_t = \mathcal{P}Q_t$ satisfies the recursion

$$\theta_{t+1} = \mathcal{PT}_t(F\theta_t, F\theta_t). \tag{21}$$

In particular, $Q_t = F\theta_t$ holds as well and when Q_0 is any function satisfying $\mathcal{P}Q_0 = \theta_0$ then Equations (21) and (1) yield the same process θ_t.

1996).

Now, our goal is to show that Theorem 4.1 can be applied to prove the convergence of the process Q_t (and hence that of θ_t).

First, let us consider the convergence of the process $\tilde{Q}_{t+1} = \mathcal{GT}_t(\tilde{Q}_t, Q)$, where $Q \in B(\mathcal{X} \times \mathcal{A})$ is such that $Q = F\theta_0$ for some $\theta_0 \in \mathbb{R}^n$. Consider the corresponding parameter space recursion $\mathcal{P}\tilde{Q}_{t+1} = \mathcal{PT}_t(F\mathcal{P}\tilde{Q}_t, F\theta_0)$. This recursion takes a form similar to multi-state Q-learning whose convergence was proven in Theorem 4 of (Szepesvári & Littman, 1999). The only difference is that here s_{ti} is defined with $s_{ti} = s(x_i, a_i, X_t)$, where now X_t is the element of the not necessarily finite set $\mathcal{X}$. Also, in Theorem 4 it was assumed that X_t is stationary. Nevertheless, the same proof applies with some small changes once we show that the conditions $\sum_{t=1}^{\infty} \alpha_{ti} = \infty$ and $\sum_{t=1}^{\infty} \alpha_{ti}^2 < \infty$ hold w.p.1. The rest of the assumptions are readily satisfied. Consequently we will have that $\mathcal{P}\tilde{Q}_t$ converges to $\mathcal{PH}F\theta_0 = \mathcal{PH}Q$, where $\mathcal{H}$ is defined by (18) (note that $\mathcal{PH}F$ is an operator over a finite dimensional vector space). Hence $\tilde{Q}_t$ converges to $\mathcal{GH}Q$ w.p.1.

The conditions on the learning rates α_{ti} are satisfied since (X_t) is positive Harris and therefore for any fixed i, $n_t(x_i, a_i)/(t + 1) \to \mu_X(A_i) > 0$.[4] The extension of Theorem 4 of (Szepesvári & Littman, 1999) to non-stationary processes can be obtained thanks to $E[f(X_t)|H_t] \to \int f(x) d\mu_X(x)$ as $t \to \infty$, where f is any function satisfying $f \in L^1(\mu)$. This convergence holds since (X_t) is positive Harris and aperiodic (cf. Theorem 13.3.3 of (Meyn et al., 1996)). This together with some trivial extensions of Theorem 7 and Lemma 7 of (Szepesvári & Littman, 1999) yield the convergence of $\mathcal{P}\tilde{Q}_t$ to $\mathcal{PH}F\theta_0$.[5]

So far we have seen that for $Q = F\theta_0$ we have that $\tilde{Q}_t \to \mathcal{GH}Q$. Now let θ^* be the fixed point of $\mathcal{PH}F$. Then $F\theta^*$ is the fixed point of $\mathcal{GH}$. Hence, if one takes θ_0 then by our previous result we get that the first condition of Theorem 4.1 is satisfied. The second condition can be checked directly using the definition of $\mathcal{T}_t$. This shows that Q_t converges to the fixed point of $\mathcal{GH}$, and hence also that θ_t converges to some parameter vector θ^* such that $F\theta^*$ is the fixed point of $\mathcal{GH}$. □

Note that by taking $|A| = 1$, Theorem 4.2 yields the

[4] This follows since according to Theorem 17.1.7 of (Meyn et al., 1996), any positive Harris recurrent chain satisfies the law of large numbers.

[5] Due to the lack of space here and in what follows we restrict ourselves to communicating the main ideas and leaving technical details to the reader and a forthcoming longer version of this paper.

convergence of $TD(0)$ for any fixed positive stochastic stationary policy π.

5. Extensions

In this section we consider several extensions of the basic algorithm. The first extension concerns adaptive basis point construction methods.

5.1. Adaptive Basis Points

Often the set of basis points is determined by means of an adaptive process. The underlying assumption is that the function approximator is more accurate where there are more basis points. This is the case when e.g. barycentric (linear) interpolation is used, or more generally for spline- or kernel-based methods. Hence, the goal of the algorithm that determines the location of the basis points is to put more basis points into regions where a more accurate representation is required (e.g. (Munos & Moore, 1999)). In the current paper, we are not concerned with the details of such a construction, but are interested in the convergence of the resulting algorithm. The only assumption we make on the construction process is that it should depend only on the past observations and that the set of basis points is changed by the process only a finite number of times. Further, we assume that the last time when the set of basis points is updated is bounded with probability one. We call this assumption (FT). The notion of function approximators need to be extended to cover the case of a variable number of basis points. This is done by assuming that we have a sequence of function approximators $F^{(n)}$ such that $F^{(n)} : \mathbb{R}^n \times Z^n \to B(\mathcal{Z})$, where we introduced $\mathcal{Z} = \mathcal{X} \times \mathcal{A}$. Further, we assume that the following error bounds hold:

$$\|F^{(n)}(\mathcal{P}Q, S) - Q\| \le C \operatorname{dens}(S), \qquad (22)$$

where $Q \in B(\mathcal{Z})$ is now assumed to be a continuous function living in an appropriate smoothness space L, such as a Lipschitz space, and where the constant $C > 0$ is independent of S and Q. We shall also assume that the image space of $F^{(n)}$ is contained in L. Here $\operatorname{dens}(S)$ refers to the density of S. This is defined as $\max_{z\in\mathcal{Z}} \operatorname{dist}(z, S)$, where $\operatorname{dist}(z, S) = \min_{s\in S} d(z, s)$, and where d is a distance defined over $\mathcal{Z}$. Let n_t denote the number of basis points at time t. The following result holds:

Proposition 5.1. *Assume that the basic algorithm is run parallel to a process that updates the set of basis points based on past observations such that in all steps the maximum number of points added is bounded by some constant. Let S_t be the set of basis points at time t. Assume that assumption (FT) holds. Then θ_t will converge w.p.1. to some random vector θ^*, such that $F\theta^*$ is the fixed point of the (random) operator $\mathcal{GH}$, where $\mathcal{G}$ is defined by $\lim_{t\to\infty} F^{(n_t)}(\cdot, S_t)\mathcal{P}$. If we also have that $\limsup_{t\to\infty} \operatorname{dens}(S_t) < h_0$ holds w.p.1 and if Q^*, the fixed point of the operator $\mathcal{H}$ is sufficiently smooth ($Q^* \in L$) and if $\mathcal{H}$ maps L into itself then $\|F\theta^* - Q^*\| \le O(h_0/(1-\gamma))$.*

Note that smoothness requirements regarding $\mathcal{H}$ and its fixed point can be satisfied when one assumes sufficient smoothness and regularity of the immediate reward function r and the transition probability kernel P, such as if r is bounded and differentiable and P satisfies the so-called Feller property.

5.2. A Multi-stage Process

In this section we consider a multi-stage process with the goal of proving convergence to the fixed point of the Bellman operator underlying the MDP. In the proposed multi-stage process the parameter update equation is used as a subroutine of an outer cycle. The purpose of the outer cycle is to increase the density of the set of basis points S so that convergence will not be limited by the denseness of this set. Let therefore S_t be a (deterministic) sequence of nested sets with $\operatorname{dens}(S_t) \to 0$. Assume that the sets are changed at the time steps $t_0, t_1, \ldots, t_n, \ldots$ where $t_{n+1} - t_n \ge 0$ and $t_{n+1} - t_n \to +\infty$ at an appropriate rate. Further, assume that we are also given a sequence of non-expansive operators $J_t : \mathbb{R}^{|S_t|} \to B(\mathcal{Z})$, where J_t is interpolative with respect to S_t. We assume that when at time t_j the set S_{t_j-1} is replaced by the set S_{t_j} the new parameters θ'_{t_j} are determined in such a way that no information is lost. Specifically, we assume that $\theta'_{t_j} = \mathcal{P}_{S_{t_j}} J_{t_j-1}\theta_{t_j}$.[6] We further assume that that (22) holds for the sequence J_t, as well.

We shall also change the definition of s_{ti} by letting the spatial smoother shrink with time and by compensating for the effect of the exploration policy π. Assuming that μ_X is absolutely continuous w.r.t. the Lebesgue measure $d\lambda$, we define $g_\pi(x) = d\mu_X/d\lambda$ as the density of μ_X. We assume again sufficient regularity (e.g. fast mixing, $g_\pi(x)$ being continuous and bounded away from zero) and introduce κ_t, a density estimator whose purpose is to estimate $g_\pi(x)$ using the samples (X_t). We redefine s_{ti} as follows:

$$s_{ti} = \frac{s_t(x_{ti}, a_{ti}, X_t)}{\kappa_t(X_t)}.$$

Here (x_{ti}, a_{ti}) are the elements of S_t and $s_t : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to \mathbb{R}$ is a sequence of functions such that $s_t(x, a, y) =$

[6] Note that this simply means that old components are kept unchanged and new components of θ'_{t_j} are set by reading out the value of $J_{t_j-1}\theta_{t_j}$ at the new site corresponding to a given new component.

$\phi_a(\|x-y\|/h_t)$ for some functions $\phi_a : \mathbb{R}_0^+ \to \mathbb{R}$ that are assumed to be continuous and satisfy $\phi_a(r) \to 0$ as $r \to \infty$ sufficiently fast. We further assume that the smoothing bandwidth h_t converges to 0. For the sake of simplicity we assume that h_t and κ_t are kept constant in the intervals $[t_0, t_1)$, $[t_1, t_2)$, ... We assume that κ_t is constructed such that it converges uniformly to g_π w.p.1. Such a density estimator can be constructed using e.g. kernel density estimators (Liebscher, 2001). The following proposition holds:

Proposition 5.2. *Under sufficient regularity assumptions on the MDP $\mathcal{M}$ the estimates θ_t are such that $J_t\theta_t$ converges to the fixed point of the Bellman operator T underlying the MDP $\mathcal{M}$.*

The proof uses uniform bounds on κ_t, a convergence rate estimate of the basic process (1) (where s_t is defined using a fixed spatial smoother, s) along the lines of (Even-Dar & Mansour, 2003) and the property that if $\mathcal{H}_s$ is defined by

$$\mathcal{H}_s(Q)(z,a) = \int\int s(z,a,x)\{r(x,a) + \gamma \max_b Q(y,b)\} dP(y|x,a) d\mu_X(x). \tag{23}$$

then for $s_t(z,a,x) = \phi_a(\|z-x\|/h_t)/g_t(x)$ with $h_t \to 0$, $g_t \to g_\pi$, $\lim_{t\to\infty} \mathcal{H}_{s_t} \to T$ holds where T is the Bellman operator of the underlying MDP. The proof is omitted due to the lack of space. Note that this result can be easily extended to the case when in each stage a different (but "proper") sampling policy is used.

6. Related Work

In (Gordon, 1995) and indepentently 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 or an 'almost non-expansion'. Both papers assume that the controlled system is known.

The only work known to us which does need a sampling device or the knowledge of a model and which does not build a model is due to Singh et al. (1995). The algorithm introduced by these authors is called "soft-state aggregation" (SSA) and works by updating the parameters θ_t much like Equation (1) except that neither a spatial smoother, nor a function approximator is used in the update equation and only one component is updated in each time step. For convenience we assume that components of θ_t are now indexed by pairs of the form (i,a), $1 \le i \le n$, $a \in A$. Then in step t only component with index (i_t, A_t) is updated with $i_t \sim P(\cdot|X_t)$, where $P(\cdot|\cdot)$ is a parameter of the algorithm. Also, $\max_b \theta_{t,(i_{t+1},b)}$ is used as the estimate of the value of the "next state" in place of $\max_b F_{\theta_t}(X_{t+1}, b)$. Finally, the value of a state-action pair is computed by interpolating among the components of θ_t using the probability distribution P: $Q_t(x,a) = \sum_i P(i|x)\theta_{t,(i,a)}$. Having said this, our algorithm can be viewed as a Rao-Blackwellised version of SSA with $F_{\theta_t}(x,a) = E[\theta_{t,(i_t,a)}|X_t = x, \theta_t]$ and $s_{ti} = P(i|X_t)$. Although, our theoretical results do not apply to this case since F_θ defined this way will not be an interpolative function approximator, intuition still says that our algorithm (with some other interpolative non-expansion) should be more efficient than SSA since it avoids the introduction of additional randomness and thus it should yield estimates having smaller variance.

Another related method is Kernel-based Reinforcement Learning (KBRL) introduced by (Ormoneit & Sen, 2002). This algorithm is best viewed as one that uses non-parametric kernel-methods to estimate the model (the reward function and the transition probabilities). Since it uses kernels and since it is a non-parametric method at a surface-level it might look similar to our algorithm, but even for a fixed set of sample points our algorithm converges to a different limit point. Further, KBRL is best viewed as an off-line algorithm, whilst our method is an on-line method.

7. Experimental Results

We have run some experiments where we compared our algorithm (henceforth called IFAPPQ for "Interpolative Function Approximator based Q-learning") with SSA and KBRL. We have selected the well-known 'mountain car' domain of Singh and Sutton (1996) because it provides a standard test-bed and thus our results can be compared with other results published in the literature.

For all the three algorithms, samples were generated using the same fixed stochastic stationary policy with each run being started at a uniform random location in the state space. This sampling policy was chosen to be an 0.2-greedy policy corresponding to a finely-tuned Q-table. Performance was measured as the L^2-error of the learned Q-values where the values of the finely-tuned Q-table were taken as the ground truth. The L^2-error was measured (approximately) only over those parts of the state space that had a subtantial chance of being sampled by our sampling method.[7]

For SSA we used a fixed number of basis points that were sampled uniformly at random. We tried vari-

[7] This part of the state space was precomputed by running a series of Monte-Carlo experiments.

ous number in the range $[50, 400]$ and finally chose to use 200. This parameter had no substantial influance on the final performance. For IFAPPQ we used the following adaptive basis point construction algorithm: when the closest basis point to a new sample point is farther away than a constant (0.05) then a new basis point is inserted and the corresponding parameter values are set such that the action-values at the inserted point do not change due to the insertion. At the end of runs we typically ended up with 280–290 basis points. All basis points stored the distance to their closest neighbors to improve efficiency. Both s and F_θ use Gaussian kernels whose bandwidth is set such that they evaluate to less than 0.01 at their closest neighbors. For F_θ we used kernel-based averaging.[8] The learning rate α_{ti} is set to be $a_0/(1+n_{ti}/N_0)$, where we used $N_0 = 1000$ and $a_0 = 1$. For KBRL we used a fixed bandwidth that matched the bandwidth used with IFAPPQ. We also tried time varying bandwidths but the results did not improve significantly.

We decided to compare the algorithms on the number of floating point operations they use. The reason is that for the same amount of data KBRL does much more computation than the other algorithms and makes better use of experience initially. However we did not compare the performance of it with that of the other algorithms on the basis of the number of samples observed because of its excessive computational demands.

Plots of the estimated L^2-error of the Q-values are shown in Figure 1 obtained by averaging results of 5 independent tests. It can be seen that the performance of KBRL is better than that of the others initially, but as time goes by the error of IFAPPQ becomes lower. For SSA the performance improves initially and then it gets worse again. We are still investigating this.

8. Conclusions

We have derived rigorous convergence results for Q-learning when combined with appropriate function approximators. In addition to being a non-expansion, we require the function approximator to satisfy an interpolation property. This result was extended to algorithms that construct the set of basis points in an adaptive manner. To our best knowledge this is the first result that concerns the convergence of such processes. The basic algorithm was also extended to a multi-stage process and it was argued that using the tools available to us this process can be shown to converge to the optimal value function of the underlying MDP in probability. In the multi-stage setting it is also possible to remove the assumption that the policy used to sample the MDP must be fixed.

[8] Although kernel-based averagers are not interpolative, they are quasi-interpolative in the sense that $F_\theta(x_i) - \theta_i$ is guaranteed to be small when the kernels' bandwidths match the density of the basis point set. Our results can be extended to such cases.

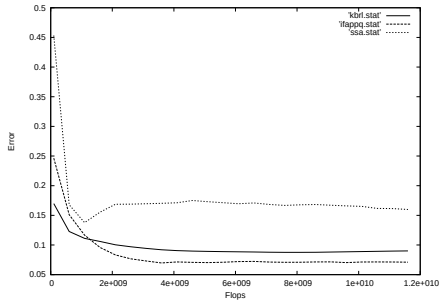

Figure 1. The figure shows the estimated L^2-approximation error of Q^* obtained as a number of floating point operations used by the various algorithms.

References

Even-Dar, E., & Mansour, Y. (2003). Learning rates for Q-learning. *Journal of Machine Learning Research*, *5*, 1–25.

Gordon, G. J. (1995). Stable function approximation in dynamic programming. *Proc. of ICML 20* (pp. 261–268). Morgan Kaufmann.

Liebscher, E. (2001). Estimation of the density and the regression function under mixing conditions. *Statistics & Decisions*, *19*, 9–26.

Meyn, S., , & Tweedie, R. (1996). *Markov chains and stochastic stability*. Springer-Verlag.

Munos, R., & Moore, A. (1999). Variable resolution discretization for high-accuracy solutions of optimal control problems. *Proc. of IJCAI* (pp. 1348–1355).

Ormoneit, D., & Sen, S. (2002). Kernel-based reinforcement learning. *Machine Learning*, *49*, 161–178.

Singh, S., Jaakkola, T., & Jordan, M. (1995). Reinforcement learning with soft state aggregation. *NIPS 7* (pp. 361–368). MIT Press.

Singh, S., & Sutton, R. (1996). Reinforcement learning with replacing eligibility traces. *Machine Learning*, *32*, 123–158.

Szepesvári, C., & Littman, M. (1999). A unified analysis of value-function-based reinforcement-learning algorithms. *Neural Computation*, *11*, 2017–2059.

Tsitsiklis, J. N., & Van Roy, B. (1996). Feature-based methods for large scale dynamic programming. *Machine Learning*, *22*, 59–94.