An Analysis of Reinforcement Learning with Function Approximation

Francisco S. Melo FMELO@CS.CMU.EDU
Carnegie Mellon University, Pittsburgh, PA 15213, USA

Sean P. Meyn MEYN@CONTROL.CSL.UIUC.EDU
Coordinated Science Lab, Urbana, IL 61801, USA

M. Isabel Ribeiro MIR@ISR.IST.UTL.PT
Institute for Systems and Robotics, 1049-001 Lisboa, Portugal

Abstract

We address the problem of computing the optimal Q-function in Markov decision problems with infinite state-space. We analyze the convergence properties of several variations of Q-learning when combined with function approximation, extending the analysis of TD-learning in (Tsitsiklis & Van Roy, 1996a) to stochastic control settings. We identify conditions under which such approximate methods converge with probability 1. We conclude with a brief discussion on the general applicability of our results and compare them with several related works.

1. Introduction

Convergence of Q-learning with function approximation has been a long standing open question in reinforcement learning (Sutton, 1999). In general, value-based reinforcement learning (RL) methods for optimal control behave poorly when combined with function approximation (Baird, 1995; Tsitsiklis & Van Roy, 1996a). In this paper, we address this problem by analyzing the convergence of Q-learning when combined with linear function approximation. We identify a set of conditions that imply the convergence of this approximation method with probability 1 (w.p.1), when a fixed learning policy is used, and provide an interpretation of the resulting approximation as the fixed point of a Bellman-like operator. This motivates the analysis of several variations of Q-learning when combined with linear function approximation. In particular, we study a variation of Q-learning using importance sampling and an on-policy variant of Q-learning (SARSA).

The paper is organized as follows. We start in Section 2 by describing Markov decision problems. We proceed with our analysis of the Q-learning algorithm and its variants, and produce our main results in Section 3. We also compare our results with other related works in the RL literature. We conclude with some further discussion in Section 4.

2. Markov Decision Problems

Let $(\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$ be a Markov decision problem (MDP) with a compact state-space $\mathcal{X} \subset \mathbb{R}^p$ and a finite action set $\mathcal{A}$. The action-dependent kernel P_a defines the transition probabilities for the underlying controlled Markov chain $\{X_t\}$ as

$$\mathbb{P}\left[X_{t+1} \in U \mid X_t = x, A_t = a\right] = \mathsf{P}_a(x, U),$$

where U is any measurable subset of $\mathcal{X}$. The $\mathcal{A}$-valued process $\{A_t\}$ represents the control process: A_t is the control action at time instant t.[1] Solving the MDP consists in determining the control process $\{A_t\}$ maximizing the expected total discounted reward

$$V\big(\{A_t\}, x\big) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(X_t, A_t) \mid X_0 = x\right],$$

where $0 \leq \gamma < 1$ is a discount-factor and $R(x, a)$ represents a random "reward" received for taking action $a \in \mathcal{A}$ in state $x \in \mathcal{X}$. For simplicity of notation, we consider a bounded deterministic function $r : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \longrightarrow \mathbb{R}$ assigning a reward $r(x, a, y)$ every time a transition from x to y occurs after taking

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008.

[1] We take the control process $\{A_t\}$ to be adapted to the σ-algebra induced by $\{X_t\}$.

action a. This means that

$$\mathbb{E}\left[R(x,a)\right] = \int_{\mathcal{X}} r(x,a,y)\mathsf{P}_a(x,dy).$$

The *optimal value function* V^* is defined for each state $x \in \mathcal{X}$ as

$$V^*(x) = \max_{\{A_t\}} V\big(\{A_t\}, x\big) = \\ = \max_{\{A_t\}} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(X_t, A_t) \mid X_0 = x\right]$$

and verifies the Bellman optimality equation

$$V^*(x) = \max_{a\in\mathcal{A}} \int_{\mathcal{X}} \big[r(x,a,y) + \gamma V^*(y)\big]\mathsf{P}_a(x,dy). \quad (1)$$

$V^*(x)$ represents the expected total discounted reward received along an optimal trajectory starting at state x. We can also define the optimal Q-function Q^* as

$$Q^*(x,a) = \int_{\mathcal{X}} \big[r(x,a,y) + \gamma V^*(y)\big]\mathsf{P}_a(x,dy), \quad (2)$$

representing the expected total discounted reward along a trajectory starting at state x obtained by choosing a as the first action and following the optimal policy thereafter. The control process $\{A_t\}$ defined as

$$A_t = \arg\max_{a\in\mathcal{A}} Q^*(X_t, a), \quad \forall t,$$

is optimal in the sense that $V(\{A_t\}, x) = V^*(x)$ and defines a mapping $\pi^* : \mathcal{X} \to \mathcal{A}$ known as the *optimal policy*. The optimal policy determines the optimal decision rule for a given MDP.

More generally, a (Markov) *policy* is any mapping π_t defined over $\mathcal{X} \times \mathcal{A}$ generating a control process $\{A_t\}$ verifying, for all t,

$$\mathbb{P}\left[A_t = a \mid X_t = x\right] = \pi_t(x,a), \quad \forall t.$$

We write $V^{\pi_t}(x)$ instead of $V(\{A_t\}, x)$ if the control process $\{A_t\}$ is generated by policy π_t. A policy π_t is *stationary* if it does not depend on t and *deterministic* if it assigns probability 1 to a single action in each state and is thus represented as a map $\pi_t : \mathcal{X} \to \mathcal{A}$ for every t. Notice that the optimal control process can be obtained from the optimal (stationary, deterministic) policy π^*, which can in turn be obtained from Q^*. Therefore, the optimal control problem is solved once the function Q^* is known for all pairs (x,a).

Now given any real function q defined over $\mathcal{X} \times \mathcal{A}$, we define the Bellman operator

$$(\mathbf{H}q)(x,a) = \int_{\mathcal{X}} \big[r(x,a,y) + \gamma \max_{u\in\mathcal{A}} q(y,u)\big]\mathsf{P}_a(x,dy). \quad (3)$$

The function Q^* in (2) is the fixed-point of $\mathbf{H}$ and, since this operator is a contraction in the sup-norm, a fixed-point iteration can be used to determine Q^* (at least theoretically).

2.1. The Q-Learning Algorithm

We previously suggested that a fixed-point iteration could be used to determine the function Q^*. In practice, this requires two important conditions:

- The kernel P and the reward function r are known;
- The successive estimates for Q^* can be represented compactly and stored in a computer with finite memory.

If P and/or r are not known, a fixed-point iteration using $\mathbf{H}$ is not possible. To solve this problem, Watkins proposed in 1989 the *Q-learning algorithm* (Watkins, 1989). Q-learning proceeds as follows: consider a MDP $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$ and suppose that $\{x_t\}$ is an infinite sample trajectory of the underlying Markov chain obtained with some policy π_t. The corresponding sample control process is denoted as $\{a_t\}$ and the sequence of obtained rewards as $\{r_t\}$. Given any initial estimate Q_0, Q-learning successively updates this estimate using the rule

$$Q_{t+1}(x,a) = Q_t(x,a) + \alpha_t(x,a)\Delta_t, \quad (4)$$

where $\{\alpha_t\}$ is a step-size sequence and Δ_t is the temporal difference at time t,

$$\Delta_t = r_t + \gamma \max_{b\in\mathcal{A}} Q_t(x_{t+1}, b) - Q_t(x_t, a_t). \quad (5)$$

If both $\mathcal{X}$ and $\mathcal{A}$ are finite sets, each estimate Q_t is simply a $|\mathcal{X}| \times |\mathcal{A}|$ matrix and can be represented explicitly in a computer. In that case, the convergence of Q-learning and several other related algorithms (such as TD(λ) or SARSA) has been thoroughly studied (see, for example, (Bertsekas & Tsitsiklis, 1996) and references therein). However, if either $\mathcal{X}$ or $\mathcal{A}$ are infinite or very large, explicitly representing each Q_t becomes infeasible and some form of compact representation is needed (*e.g.*, using function approximation). In this paper, we address how several RL methods such as Q-learning and SARSA can be combined with function approximation and still retain their main convergence properties.

3. Reinforcement Learning with Linear Function Approximation

In this section, we address the problem of determining the optimal Q-function for MDPs with infinite state-space $\mathcal{X}$. Let $\mathcal{Q} = \{Q_\theta\}$ be a family of real-valued

functions defined in $\mathcal{X} \times \mathcal{A}$. It is assumed that the function class is linearly parameterized, so that $\mathcal{Q}$ can be expressed as the linear span of a *fixed set* of M linearly independent functions $\phi_i : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$. For each M-dimensional parameter vector $\theta \in \mathbb{R}^M$, the function $Q_\theta \in \mathcal{Q}$ is defined as,

$$Q_\theta(x,a) = \sum_{i=1}^{M} \phi_i(x,a)\theta(i) = \phi^\top(x,a)\theta,$$

where $^\top$ represents the transpose operator. We will also denote the above function by $Q(\theta)$ to emphasize the dependence on θ over the dependency on (x,a).

Let π be a fixed stochastic, stationary policy and suppose that $\{x_t\}$, $\{a_t\}$ and $\{r_t\}$ are sampled trajectories of states, actions and rewards obtained from the MDP using policy π. In the original Q-learning algorithm, the Q-values are updated according to (4). The temporal difference Δ_t can be interpreted as a 1-step estimation error with respect to the optimal function Q^*. The update rule in Q-learning "moves" the estimates Q_t closer to the desired function Q^*, minimizing the expected value of Δ_t.

In our approximate setting, we apply the same underlying idea to obtain the update rule for approximate Q-learning:

$$\begin{aligned} \theta_{t+1} &= \theta_t + \alpha_t \nabla_\theta Q_\theta(x_t,a_t)\Delta_t \\ &= \theta_t + \alpha_t \phi(x_t,a_t)\Delta_t, \end{aligned} \tag{6}$$

where, as above, Δ_t is the temporal difference at time t defined in (5). Notice that (6) updates θ_t using the temporal difference Δ_t as the error. The gradient $\nabla_\theta Q_\theta$ provides the "direction" in which this update is performed.

To establish convergence of the algorithm (6) we adopt an ODE argument, establishing the trajectories of the algorithm to closely follow those of an associated ODE with a globally asymptotically stable equilibrium point. This will require several regularity properties on the policy π and on its induced Markov chain that will, in turn, motivate the study of the on-policy version of the algorithm. This on-policy algorithm can be seen as an extension of SARSA to infinite settings.

3.1. Convergence of Q-Learning

We now proceed by identifying conditions that ensure the convergence of Q-learning with linear function approximation as described by (6). Due to space limitations, we overlook some of the technical details in the proofs that can easily be filled in.

We start by introducing some notation that will greatly simplify the presentation. Given an MDP $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$ with compact state space $\mathcal{X} \subset \mathbb{R}^p$, let $(\mathcal{X}, \mathsf{P}_\pi)$ be the Markov chain induced by a fixed policy π. We assume the chain $(\mathcal{X}, \mathsf{P}_\pi)$ to be uniformly ergodic with invariant probability measure μ_X and the policy π to verify $\pi(x,a) > 0$ for all $a \in \mathcal{A}$ and μ_X-almost all $x \in \mathcal{X}$. We denote by μ_π the probability measure defined for each measurable set $U \subset \mathcal{X}$ and each action $a \in \mathcal{A}$ as

$$\mu_\pi(U \times \{a\}) = \int_U \pi(x,a)\mu_X(dx).$$

Let now $\{\phi_i, i = 1, \ldots, M\}$ be a set of bounded, linearly independent basis functions to be used in our approximate Q-learning algorithm. We denote by Σ_π the matrix defined as

$$\Sigma_\pi = \mathbb{E}_\pi\left[\phi(x,a)\phi^\top(x,a)\right] = \int_{\mathcal{X}\times\mathcal{A}} \phi\,\phi^\top d\mu_\pi$$

Notice that the above expression is well-defined and independent of the initial distribution for the chain, due to our assumption of uniform ergodicity.

For fixed $\theta \in \mathbb{R}^M$ and $x \in \mathcal{X}$, define the set of maximizing actions at x as

$$\mathcal{A}_x^\theta = \{a^* \in \mathcal{A} \mid \phi^\top(x,a^*)\theta = \max_a \phi^\top(x,a)\theta\}$$

and the greedy policy with respect to θ as any policy that, at each state x, assigns positive probability only to actions in $\mathcal{A}_x^\theta$. Finally, let ϕ_x denote the row-vector $\phi^\top(x,a)$, where a is a random action generated according to the policy π at state x; likewise, let ϕ_x^θ denote the row-vector $\phi^\top(x,a_x^\theta)$, where a_x^θ is now any action in $\mathcal{A}_x^\theta$. We now introduce the *θ-dependent* matrix

$$\Sigma_\pi^*(\theta) = \mathbb{E}_\pi\left[\left(\phi_x^\theta\right)^\top \phi_x^\theta\right].$$

By construction, both Σ_π and Σ_π^* are positive definite, since the functions ϕ_i are assumed linearly independent. Notice also the difference between Σ_π and each Σ_π^*: the actions in the definition of the former are taken according to π while in the latter they are taken greedily with respect to a particular θ.

We are now in position to introduce our first result.

Theorem 1 *Let $\mathcal{M}$, π and $\{\phi_i, i = 1, \ldots, M\}$ be as defined above. If, for all θ,*

$$\Sigma_\pi > \gamma^2 \Sigma_\pi^*(\theta) \tag{7}$$

and the step-size sequence verifies

$$\sum_t \alpha_t = \infty \qquad \sum_t \alpha_t^2 < \infty,$$

then the algorithm in (6) *converges w.p.1 and the limit point* θ^* *verifies the recursive relation*

$$Q(\theta^*) = \Pi_{\mathcal{Q}} \mathbf{H} Q(\theta^*),$$

where $\Pi_{\mathcal{Q}}$ *is the orthogonal projection onto* $\mathcal{Q}$.[2]

PROOF We establish the main statement of the theorem using a standard ODE argument.

The assumptions on the chain $(\mathcal{X}, \mathsf{P}_\pi)$ and basis functions $\{\phi_i, i = 1, \ldots, M\}$ and the fact that $\pi(x, a) > 0$ for all $a \in \mathcal{A}$ and μ_X-almost every $x \in \mathcal{X}$ ensure the applicability of Theorem 17 in page 239 of (Benveniste et al., 1990). Therefore, the convergence of the algorithm can be analyzed in terms of the stability of the equilibrium points of the associated ODE

$$\dot{\theta} = \mathbb{E}_\pi \Big[\phi_x^\top \big(r(x, a, y) + \gamma \phi_y^\theta \theta - \phi_x \theta \big) \Big], \qquad (8)$$

where we omitted the explicit dependence of θ on t to avoid excessively cluttering the expression. If the ODE (8) has a globally asymptotically stable equilibrium point, this implies the algorithm (6) to converge w.p.1 (Benveniste et al., 1990). Let then $\theta_1(t)$ and $\theta_2(t)$ be two trajectories of the ODE starting at different initial conditions, and let $\tilde{\theta}(t) = \theta_1(t) - \theta_2(t)$. From (8), we get

$$\frac{d}{dt} \|\tilde{\theta}\|_2^2 = -2 \tilde{\theta}^\top \Sigma_\pi \tilde{\theta} + 2\gamma \mathbb{E}_\pi \left[\big(\phi_x \tilde{\theta} \big) \big(\phi_y^{\theta_1} \theta_1 - \phi_y^{\theta_2} \theta_2 \big) \right].$$

Notice now that, from the definition of $\phi_y^{\theta_1}$ and $\phi_y^{\theta_2}$,

$$\phi_y^{\theta_1} \theta_2 \leq \phi_y^{\theta_2} \theta_2 \qquad \phi_y^{\theta_2} \theta_1 \leq \phi_y^{\theta_1} \theta_1.$$

Taking this into account and defining the sets $S_+ = \{(x, a) \mid \phi^\top(x, a) \tilde{\theta} > 0\}$ and $S_- = \mathcal{X} \times \mathcal{A} - S_+$, the previous expression becomes

$$\frac{d}{dt} \|\tilde{\theta}\|_2^2 \leq -2 \tilde{\theta}^\top \Sigma_\pi \tilde{\theta} + 2\gamma \mathbb{E}_\pi \left[\big(\phi_x \tilde{\theta} \big) \big(\phi_y^{\theta_1} \tilde{\theta} \big) \mathbb{I}_{S_+} \right] + 2\gamma \mathbb{E}_\pi \left[\big(\phi_x \tilde{\theta} \big) \big(\phi_y^{\theta_2} \tilde{\theta} \big) \mathbb{I}_{S_-} \right],$$

where $\mathbb{I}_S$ represents the indicator function for the set S. Applying Hölder's inequality to each of the expectations above, we get

$$\frac{d}{dt} \|\tilde{\theta}\|_2^2 \leq -2 \tilde{\theta}^\top \Sigma_\pi \tilde{\theta} + 2\gamma \sqrt{\mathbb{E}_\pi \left[(\phi_x \tilde{\theta})^2 \mathbb{I}_{S_+} \right] \mathbb{E}_\pi \left[(\phi_x^{\theta_1} \tilde{\theta})^2 \mathbb{I}_{S_+} \right]} + 2\gamma \sqrt{\mathbb{E}_\pi \left[(\phi_x \tilde{\theta})^2 \mathbb{I}_{S_-} \right] \mathbb{E}_\pi \left[(\phi_x^{\theta_2} \tilde{\theta})^2 \mathbb{I}_{S_-} \right]}$$

and a few simple computations finally yield

$$\frac{d}{dt} \|\tilde{\theta}\|_2^2 \leq -2 \tilde{\theta}^\top \Sigma_\pi \tilde{\theta} + 2\gamma \sqrt{\tilde{\theta}^\top \Sigma_\pi \tilde{\theta} \max \big(\tilde{\theta}^\top \Sigma_\pi^*(\theta_1) \tilde{\theta}, \tilde{\theta}^\top \Sigma_\pi^*(\theta_2) \tilde{\theta} \big)}.$$

Since, by assumption, $\Sigma_\pi > \gamma^2 \Sigma_\pi^*(\theta)$, we can conclude from the expression above that

$$\frac{d}{dt} \|\tilde{\theta}\|_2^2 < 0.$$

This means, in particular, that $\tilde{\theta}(t)$ converges asymptotically to the origin, *i.e.*, the ODE (8) is globally asymptotically stable. Since the ODE is autonomous (*i.e.*, time-invariant), there exists one globally asymptotically stable equilibrium point for the ODE, that verifies the recursive relation

$$\theta^* = \Sigma_\pi^{-1} \mathbb{E}_\pi \left[\phi_x \big(r(x, a, y) + \gamma \phi_y^{\theta^*} \theta^* \big) \right]. \qquad (9)$$

Since Σ_π is, by construction, positive definite, the inverse in (9) is well-defined. Multiplying (9) by $\phi^\top(x, a)$ on both sides yields the desired result. □

It is now important to observe that condition (7) is quite restrictive: since γ is usually taken close to 1, condition (7) essentially requires that, for every θ,

$$\max_{a \in \mathcal{A}} \phi^\top(x, a) \theta \approx \sum_{a \in \mathcal{A}} \pi(x, a) \phi^\top(x, a) \theta.$$

Therefore, such condition will seldom be met in practice, since it implies that the learning policy π is already close to the policy that the algorithm is meant to compute. In other words, the maximization above yields a policy close to the policy used during learning. And, when this is the case, the algorithm essentially behaves like an *on-policy algorithm.*

On the other hand, the above condition can be ensured by considering only a local maximization around the learning policy π. This is the most interesting aspect of the above result: it explicitly relates how much information the learning policy provides about greedy policies, as a function of γ. To better understand this,

[2] The orthogonal projection is naturally defined in the (infinite-dimensional) Hilbert space containing $\mathcal{Q}$ with inner-product given by

$$\langle f, g \rangle = \int_{\mathcal{X} \times \mathcal{A}} f \, g \, d\mu_\pi.$$

notice that each policy π is associated with a particular invariant measure on the induced chain $(\mathcal{X}, \mathsf{P}_\pi)$. In particular, the measure associated with the learning policy may be very different from the one induced by the greedy/optimal policy. Taking into account the fact that γ measures, in a sense, the "importance of the future", Theorem 1 basically states that:

In problems where the performance of the agent greatly depends on future rewards ($\gamma \approx 1$), the information provided by the learning policy can only be "safely generalized" to nearby greedy policies, in the sense of (7). In problems where the performance of the agent is less dependent on future rewards ($\gamma \ll 1$), the information provided by the learning policy can be safely generalized to more general greedy policies.

Suppose then that the maximization in the update equation (6) is to be replaced by a local maximization. In other words, instead of maximizing over all actions in $\mathcal{A}$, the algorithm should maximize over a small neighborhood of the learning policy π (in policy space). The difficulty with this approach is that such maximization can be hard to implement. The use of *importance sampling* can readily overcome such difficulty, by making the maximization in policy-space *implicit*. The algorithm thus obtained, which resembles in many aspects the one proposed in (Precup et al., 2001), is described by the update rule

$$\theta_{t+1} = \theta_t + \alpha_t \phi(x_t, a_t) \hat{\Delta}_t, \tag{10}$$

where the modified temporal difference $\hat{\Delta}_t$ is given by

$$\hat{\Delta}_t = r_t + \gamma \sum_b \frac{\pi_\theta(x_{t+1}, b)}{\pi(x_{t+1}, b)} Q_{\theta_t}(x_{t+1}, b) - Q_{\theta_t}(x_t, a_t), \tag{11}$$

where π_θ is, for example, a θ-dependent ε-greedy policy close to the learning policy π.[3] A possible implementation of such algorithm is sketched in Figure 1. We denoted by π_N the behavior policy at iteration N of the algorithm; in the stopping condition for the algorithm, any adequate policy norm can be used.

3.2. SARSA with Linear Function Approximation

The analysis in the previous subsection suggests that on-policy algorithms may potentially yield more reliable convergence properties. Such fact has already been observed in (Tsitsiklis & Van Roy, 1996a; Perkins & Pendrith, 2002). In this subsection we thus focus on on-policy algorithms. We analyze the convergence of SARSA when combined with linear function approximation. In our main result, we recover the essence of the result in (Perkins & Precup, 2003), although in a somewhat different setting. The main differences between our work and that in (Perkins & Precup, 2003) are discussed further ahead.

[3] Given $\varepsilon > 0$, a policy π is *ε-greedy* with respect to a function $Q_\theta \in \mathcal{Q}$ if, at each state $x \in \mathcal{X}$, it chooses a random action with probability ε and a greedy action $a \in \mathcal{A}_x^\theta$ with probability $(1 - \varepsilon)$.

Algorithm 1 Modified Q-learning.

Require: Initial policy π_0;
1: Initialize $X_0 = x_0$ and set $N = 0$;
2: **for** $t = 0$ until T **do**
3: Sample $A_t \sim \pi_N(x_t, \cdot)$;
4: Sample next-state $X_{t+1} \sim \mathsf{P}_{a_t}(x_t, \cdot)$;
5: $r_t = r(x_t, a_t, x_{t+1})$;
6: Update θ_t according to (10);
7: **end for**
8: Set $\pi_{N+1}(x, a) = \pi_{\theta^*}(x, a)$;
9: $N = N + 1$;
10: **if** $\|\pi_N - \pi_{N-1}\|$ **then**
11: **return** π_N;
12: **else**
13: Goto 2;
14: **end if**

Once again, we consider a family $\mathcal{Q}$ of real-valued functions, the linear span of a fixed set of M linearly independent functions $\phi_i : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$, and derive an on-policy algorithm to compute a parameter vector θ^* such that $\phi^\top(x, a)\theta^*$ approximates the optimal Q-function. To this purpose, and unlike what has been done so far, we now consider a *θ-dependent* learning policy π_θ verifying $\pi_\theta(x, a) > 0$ for all θ. In particular, we consider at each time step a learning policy π_{θ_t} that is ε-greedy with respect to $\phi^\top(x, a)\theta_t$ and Lipschitz continuous with respect to θ, with Lipschitz constant C (with respect to some preferred metric). We further assume that, for every fixed θ, the Markov chain $(\mathcal{X}, \mathsf{P}_\theta)$ induced by such policy is uniformly ergodic.

Let then $\{x_t\}$, $\{a_t\}$ and $\{r_t\}$ be sampled trajectories of states, actions and rewards obtained from the MDP $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$ using (at each time-step) the θ-dependent policy π_{θ_t}. The update rule for our approximate SARSA algorithm is:

$$\theta_{t+1} = \theta_t + \alpha_t \phi(x_t, a_t) \Delta_t, \tag{12}$$

where Δ_t is the temporal difference at time t,

$$\Delta_t = r_t + \gamma \phi^\top(x_{t+1}, a_{t+1})\theta_t - \phi^\top(x_t, a_t)\theta_t.$$

In order to use the SARSA algorithm above to approximate the optimal Q-function, it is necessary to slowly decay the exploration rate, ε, to zero, while guaran-

teeing the learning policy to verify the necessary regularity conditions (namely, Lipschitz continuous w.r.t. θ). However, as will soon become apparent, decreasing the exploration rate to zero will render our convergent result (and other related results) not applicable.

We are now in position to introduce our main result.

Theorem 2 *Let $\mathcal{M}$, π_{θ_t} and $\{\phi_i, i = 1, \ldots, M\}$ be as defined above. Let C be the Lipschitz constant of the learning policy π_θ with respect to θ. Assume that the step-size sequence verifies*

$$\sum_t \alpha_t = \infty \qquad \sum_t \alpha_t^2 < \infty.$$

Then, there is $C_0 > 0$ such that, if $C < C_0$, the algorithm in (12) *converges w.p.1.*

Proof We again use an ODE argument to establish the statement of the theorem.

As before, the assumptions on $(\mathcal{X}, \mathsf{P}_\theta)$ and basis functions $\{\phi_i, i = 1, \ldots, M\}$ and the fact that the learning policy is Lipschitz continuous with respect to θ and verifies $\pi(x, a) > 0$ ensure the applicability of Theorem 17 in page 239 of (Benveniste et al., 1990). Therefore, the convergence of the algorithm can be analyzed in terms of the stability of the associated ODE:

$$\dot{\theta} = \mathbb{E}_\theta\Big[\phi_x^\top \big(r(x, a, y) + \gamma\phi_y\theta - \phi_x\theta\big)\Big]. \qquad (13)$$

Notice that the expectation is taken with respect to the invariant measure of the chain and learning policy, both θ-dependent. To establish global asymptotic stability, we re-write (13) as

$$\dot{\theta}(t) = \mathbf{A}_\theta\theta(t) + \mathbf{b}_\theta$$

where

$$\mathbf{A}_\theta = \mathbb{E}_\theta\left[\phi_x^\top\big(\gamma\phi_y - \phi_x\big)\right]; \quad \mathbf{b}_\theta = \mathbb{E}_\theta\left[\phi_x^\top r(x, a, y)\right].$$

An equilibrium point of (13) must verify $\theta^* = \mathbf{A}_{\theta^*}^{-1}\mathbf{b}_{\theta^*}$ and the existence of such equilibrium point has been established in (de Farias & Van Roy, 2000) (Theorem 5.1). Let $\tilde{\theta}(t) = \theta(t) - \theta^*$. Then,

$$\frac{d}{dt}\|\tilde{\theta}\|_2^2 = 2\tilde{\theta}^\top\big(\mathbf{A}_\theta\theta + \mathbf{b}_\theta\big) = \\ = 2\tilde{\theta}^\top\big(\mathbf{A}_\theta\theta - \mathbf{A}_{\theta^*}\theta^* + \mathbf{b}_\theta - \mathbf{b}_{\theta^*}\big).$$

Let

$$\lambda_A = \sup_\theta \|\mathbf{A}_\theta - \mathbf{A}_{\theta^*}\|_2 \quad \lambda_b = \sup_{\theta\,\neq\theta^*} \frac{\|\mathbf{b}_\theta - \mathbf{b}_{\theta^*}\|_2}{\|\theta - \theta^*\|_2},$$

where the norm in the definition of λ_A is the induced operator norm and the one in the definition of λ_b is the regular Euclidian norm. The previous expression thus becomes

$$\frac{d}{dt}\|\tilde{\theta}\|_2^2 = \\ = 2\tilde{\theta}^\top\mathbf{A}_{\theta^*}\tilde{\theta} + 2\tilde{\theta}^\top(\mathbf{A}_{\theta^*} - \mathbf{A}_\theta)\theta + 2\tilde{\theta}^\top(\mathbf{b}_\theta - \mathbf{b}_{\theta^*}) \\ \leq 2\tilde{\theta}^\top\mathbf{A}_{\theta^*}\tilde{\theta} + 2(\lambda_A + \lambda_b)\|\tilde{\theta}\|_2^2.$$

Letting $\lambda = \lambda_A + \lambda_b$, the above expression can be written as

$$\frac{d}{dt}\|\tilde{\theta}\|_2^2 \leq \tilde{\theta}^\top\big(\mathbf{A}_{\theta^*} + \lambda\mathbf{I}\big)\tilde{\theta}.$$

The fact that the learning policy is assumed Lipschitz w.r.t. θ and the uniform ergodicity of the corresponding induced chain implies that $\mathbf{A}_\theta$ and $\mathbf{b}_\theta$ are also Lipschitz w.r.t. θ (with a different constant). This means that λ goes to zero with C and, therefore, for C sufficiently small, $(A + \lambda\mathbf{I})$ is a negative definite matrix.[4] Therefore, the ODE (13) is globally asymptotically stable and the conclusion of the theorem follows. □

Several remarks are now in order. First of all, Theorem 2 basically states that, for fixed ε if the dependence of the learning policy π_θ can be made sufficiently "smooth", then SARSA converges w.p.1. This result is similar to the result in (Perkins & Precup, 2003), although the algorithms are not exactly similar: we consider a continuing task, while the algorithm featured in (Perkins & Precup, 2003) is implemented in an episodic fashion. Furthermore, in our case, convergence was established using an ODE argument, instead of the contraction argument in (Perkins & Precup, 2003). Nevertheless, both methods of proof are, in its essence, equivalent and the results in both papers concordant.

A second remark is related with the implementation of SARSA with a decaying exploration policy. The analysis of one such algorithm could be conducted using, once again, an ODE argument. In particular, SARSA could be described as a two-time-scale algorithm: the iterations of the main algorithm (corresponding to (12)) would develop on a faster time-scale and the decaying exploration rate would develop at a slower time-scale. The analysis in (Borkar, 1997) could then be replicated. However, it is well-known that, as ε approaches zero, the learning policy will approach the greedy policy w.r.t. θ which is, in general, discontinuous. Therefore, there is little hope that the smoothness

[4] The fact that $\mathbf{A}_\theta$ is negative definite has been established in several works. See, for example, Lemma 3 in (Perkins & Precup, 2003) or, in a slightly different setting (easily extendable to our setting) the proof of Theorem 1 in (Tsitsiklis & Van Roy, 1996a).

condition in Theorem 2 (or its equivalent in (Perkins & Precup, 2003)) can be met as ε approaches to zero.

4. Discussion

We now briefly discuss some of the assumptions in the above theorems.

We start by emphasizing that all stated conditions are only *sufficient*, meaning that it is possible that convergence may occur even if some (or all) fail to hold. We also discuss the relation between our results and other related works from the literature.

Seconsly, uniform ergodicity of a Markov chain essentially means that the chain quickly converges to a stationary behavior uniformly over the state-space and we can study any properties of the stationary chain by direct sampling.[5] This property and the requirement that $\pi(x,a) > 0$ for all $a \in \mathcal{A}$ and μ_X-almost all $x \in \mathcal{X}$ can be interpreted as a continuous counterpart to the usual condition that all state-action pairs are visited infinitely often. In fact, uniform ergodicity implies that all the regions of the state-space with positive μ_X measure are "sufficiently" visited (Meyn & Tweedie, 1993), and the condition $\pi(x,a) > 0$ ensures that, at each state, every action is "sufficiently" tried. It appears to be a standard requirement in this continuous scenario, as it has also been used in other works (Tsitsiklis & Van Roy, 1996a; Singh et al., 1994; Perkins & Precup, 2003).[6] The requirement that $\pi(x,a) > 0$ for all $a \in \mathcal{A}$ and μ_X-almost all $x \in \mathcal{X}$ also corresponds to the concept of *fair control* as introduced in (Borkar, 2000).

We also remark that the divergence example in (Gordon, 1996) is due to the fact that the learning policy fails to verify the Lipschitz continuity condition stated in Theorem 2 (as discussed in (Perkins & Pendrith, 2002)).

4.1. Related Work

In this paper, we analyzed how RL algorithms can be combined with linear function approximation to approximate the optimal Q-function in MDPs with infinite state-spaces. In the last decade or so, several authors have addressed this same problem from different perspectives. We now briefly discuss several such approaches and their relation to the results in this paper.

One possible approach is to rely on *soft-state aggregation* (Singh et al., 1994; Gordon, 1995; Tsitsiklis & Van Roy, 1996b), partitioning the state-space into "soft" regions. Treating the soft-regions as "hyper-states", these methods then use standard learning methods (such as Q-learning or SARSA) to approximate the optimal Q-function. The main differences between such methods and those using linear function approximation (such as the ones portrayed here) are that, in the former, only one component of the parameter vector is updated at each iteration and the basis functions are, by construction, restrained to be positive and to add to one at each point of the state-space.

Sample-based methods (Ormoneit & Sen, 2002; Szepesvári & Smart, 2004) further generalize the applicability of soft-state aggregation methods by using *spreading functions/kernels* (Ribeiro & Szepesvári, 1996). Sample-based methods thus exhibit superior convergence rate when compared with simple soft-state aggregation methods, although under somewhat more restrictive conditions.

Finally, RL with general linear function approximation was thorougly studied in (Tsitsiklis & Van Roy, 1996a; Tadić, 2001). Posterior works extended the applicability of such results. In Precup01icml, an off-policy convergent algorithm was proposed that uses an importance-sampling principle similar to the one described in Section 3. In (Perkins & Precup, 2003), the authors establish the convergence of SARSA with linear function approximation.

4.2. Concluding Remarks

We conclude by observing that all methods analyzed here as well as those surveyed above experience a degradation in performance as the distance between the target function and the chosen linear space increases. If the functions in the chosen linear space provide only a poor approximation of the desired function, there are no practical guarantees on the usefulness of such approximation. The error bounds derived in (Tsitsiklis & Van Roy, 1996a) are reassuring in that they state that the performance of approximate TD "gracefully" degrades as the distance between the target function and the chosen linear space increases. Although we have not addressed such topic in our analysis, we expect the error bounds in (Tsitsiklis & Van Roy, 1996a) to carry with little changes to our setting. Finally, we make no use of eligibility traces in our algorithms. However, it is just expectable that the methods described herein can easily be adapted to ac-

[5] Explicit bounds on the rate of convergence to stationarity are available in the literature (Meyn & Tweedie, 1994; Diaconis & Saloff-Coste, 1996; Rosenthal, 2002). However, for general chains, such bounds tend to be loose.

[6] Most of these works make use of geometric ergodicity which, since we admit a compact state-space, is a consequence of uniform ergodicity.

commodate for eligibility traces, this eventually yielding better approximations (with tighter error bounds) (Tsitsiklis & Van Roy, 1996a)

Acknowledgements

The authors would like to acknowledge the many useful comments from Doina Precup and the unknown reviewers. Francisco S. Melo acknowledges the support from the ICTI and the Portuguese FCT, under the Carnegie Mellon-Portugal Program. Sean Meyn gratefully acknowledges the financial support from the National Science Foundation (ECS-0523620). Isabel Ribeiro was supported by the FCT in the frame of ISR-Lisboa pluriannual funding. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

Baird, L. (1995). Residual algorithms: Reinforcement learning with function approximation. *Proc. 12th Int. Conf. Machine Learning* (pp. 30–37).

Benveniste, A., Métivier, M., & Priouret, P. (1990). *Adaptive algorithms and stochastic approximations*, vol. 22. Springer-Verlag.

Bertsekas, D., & Tsitsiklis, J. (1996). *Neuro-dynamic programming*. Athena Scientific.

Borkar, V. (1997). Stochastic approximation with two time scales. *Systems & Control Letters*, *29*, 291–294.

Borkar, V. (2000). A learning algorithm for discrete-time stochastic control. *Probability in the Engineering and Informational Sciences*, *14*, 243–258.

de Farias, D., & Van Roy, B. (2000). On the existence of fixed points for approximate value iteration and temporal-difference learning. *Journal of Optimization Theory and Applications*, *105*, 589–608.

Diaconis, P., & Saloff-Coste, L. (1996). Logarithmic Sobolev inequalities for finite Markov chains. *Annals of Applied Probability*, *6*, 695–750.

Gordon, G. (1995). *Stable function approximation in dynamic programming* (Technical Report CMU-CS-95-103). School of Computer Science, Carnegie Mellon University.

Gordon, G. (1996). *Chattering in SARSA(λ)*. (Technical Report). CMU Learning Lab Internal Report.

Meyn, S., & Tweedie, R. (1993). *Markov chains and stochastic stability*. Springer-Verlag.

Meyn, S., & Tweedie, R. (1994). Computable bounds for geometric convergence rates of Markov chains. *Annals of Applied Probability*, *4*, 981–1011.

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

Perkins, T., & Pendrith, M. (2002). On the existence of fixed-points for Q-learning and SARSA in partially observable domains. *Proc. 19th Int. Conf. Machine Learning* (pp. 490–497).

Perkins, T., & Precup, D. (2003). A convergent form of approximate policy iteration. *Adv. Neural Information Proc. Systems* (pp. 1595–1602).

Precup, D., Sutton, R., & Dasgupta, S. (2001). Off-policy temporal-difference learning with function approximation. *Proc. 18th Int. Conf. Machine Learning* (pp. 417–424).

Ribeiro, C., & Szepesvári, C. (1996). Q-learning combined with spreading: Convergence and results. *Proc. ISRF-IEE Int. Conf. Intelligent and Cognitive Systems* (pp. 32–36).

Rosenthal, J. (2002). Quantitative convergence rates of Markov chains: A simple account. *Electronic Communications in Probability*, *7*, 123–128.

Singh, S., Jaakkola, T., & Jordan, M. (1994). Reinforcement learning with soft state aggregation. *Adv. Neural Information Proc. Systems* (pp. 361–368).

Sutton, R. (1999). Open theoretical questions in reinforcement learning. *Lecture Notes in Computer Science*, *1572*, 11–17.

Szepesvári, C., & Smart, W. (2004). Interpolation-based Q-learning. *Proc. 21st Int. Conf. Machine learning* (pp. 100–107).

Tadić, V. (2001). On the convergence of temporal-difference learning with linear function approximation. *Machine Learning*, *42*, 241–267.

Tsitsiklis, J., & Van Roy, B. (1996a). An analysis of temporal-difference learning with function approximation. *IEEE Trans. Automatic Control*, *42*, 674–690.

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

Watkins, C. (1989). *Learning from delayed rewards*. Doctoral dissertation, King's College, University of Cambridge.