Q-learning with Uniformly Bounded Variance

Adithya M. Devraj, and Sean P. Meyn, *Fellow, IEEE*

***Abstract*—Sample complexity bounds are a common performance metric in the Reinforcement Learning literature. In the discounted cost, infinite horizon setting, all of the known bounds can be arbitrarily large, as the discount factor approaches unity. These results seem to imply that a very large number of samples is required to achieve an epsilon-optimal policy. The objective of the present work is to introduce a new class of algorithms that have sample complexity *uniformly bounded over all discount factors*. One may argue that this is impossible, due to a recent min-max lower bound. The explanation is that these prior bounds concern value function approximation and not policy approximation. We show that the asymptotic covariance of the tabular Q-learning algorithm with an optimized step-size sequence is a quadratic function of a factor that goes to infinity, as discount factor approaches 1; an essentially known result. The new *relative Q-learning* algorithm proposed here is shown to have asymptotic covariance that is uniformly bounded over all discount factors.**

***Index Terms*—Reinforcement learning, Q-learning, stochastic optimal control**

I. Introduction

MANY Reinforcement Learning (RL) algorithms can be cast as parameter estimation techniques, where the goal is to recursively estimate the parameter vector $\theta^* \in \mathbb{R}^d$ that directly, or indirectly yields an optimal decision making rule within a parameterized family. In these algorithms, the update equation for the d-dimensional parameter estimates $\{\theta_n : n \geq 0\}$ can be expressed in the general form

$$\theta_{n+1} = \theta_n + \alpha_{n+1}[\overline{f}(\theta_n) + \Delta_{n+1}], \quad n \geq 0 \tag{1}$$

in which $\theta_0 \in \mathbb{R}^d$ is given, $\{\alpha_n\}$ is a positive scalar *gain sequence* (also known as *learning rate*), $\overline{f} : \mathbb{R}^d \to \mathbb{R}^d$ is a deterministic function, and $\{\Delta_n\}$ is a "noise" sequence.

The recursion (1) is an example of stochastic approximation (SA), for which there is a vast research literature. Under mild assumptions, the estimates converge to a limit satisfying $\overline{f}(\theta^*) = 0$. Moreover, the best algorithms achieve the optimal mean-square error (MSE) convergence rate:

$$\mathsf{E}\big[\|\theta_n - \theta^*\|^2\big] = O(1/n) \tag{2}$$

It is known that TD- and Q-learning can be written in the form (1) [1], [2], [3]. In these algorithms, $\{\theta_n\}$ represents the sequence of parameter estimates that are used to approximate a *value function* or *Q-function*.

It was first established in our work [4], [3] that the convergence rate of the MSE of Watkins' Q-learning (i.e., Q-learning with a tabular basis) can be as slow as $O(1/n^{2(1-\gamma)})$, if the discount factor $\gamma \in (0, 1)$ satisfies $\gamma > \frac{1}{2}$, and if the step-size α_n is either of two standard forms (see discussion in Section IV-A). It was also shown that the optimal convergence rate (2) is obtained by using a step-size of the form $\alpha_n = g/n$, where g is a scalar proportional to $1/(1-\gamma)$; this is consistent with conclusions in more recent research [5], [6]. In the earlier work [7], a sample path *upper bound* was obtained on the rate of convergence that is roughly consistent with the mean-square rate established for $\gamma > \frac{1}{2}$ in [4], [3].

Since the publication of [7], many papers have appeared with proposed improvements to the algorithm; often including (non-asymptotic) finite-n bounds on the MSE (2). Ignoring higher order terms, these bounds can be expressed in the following general form [8], [5], [9], [6], [10]:

$$\mathsf{E}\big[\|\theta_n - \theta^*\|^2\big] \leq \frac{1}{(1-\gamma)^p} \cdot \frac{B}{n} \tag{3}$$

where $p \geq 2$ is a scalar. The constant B is a function of the total number of state-action pairs, the discount factor γ, and the maximum per-time-step cost. Much of the literature has worked towards minimizing p through a combination of hard analysis and algorithm design.

It is widely observed that Q-learning algorithms can be very slow to converge, especially when the discount factor is close to 1; The bound in (3) offers an explanation for this phenomenon. Quoting [8], a primary reason for slow convergence is "*the fact that the Bellman operator propagates information throughout the whole space*", especially when the discount factor is close to 1. We do not dispute these explanations, but in this paper we show that the challenge presented by large discounting is relatively minor. In order to make this point clear we must take a step back and rethink fundamentals: *Why do we need to estimate the Q-function?*

Letting $Q^*(x, u)$ denote the optimal Q-function evaluated at the state-action pair (x, u), the main reason for estimating the Q-function is to obtain the optimal policy:

$$\phi^*(x) := \arg\min_u Q^*(x, u)$$

It is clear from the above definition that adding a constant to Q^* *will not* alter ϕ^*. This is a fortunate fact: the Q-function can be decomposed as (see for example [11], [12], [13]):

$$Q^*(x, u) = \widetilde{Q}^*(x, u) + \frac{\eta^*}{1-\gamma} \tag{4}$$

where the scalar η^* denotes the optimal average cost, and $\widetilde{Q}^*(x, u)$ is uniformly bounded in γ, x, and u.

The reason for slow performance of Q-learning when $\gamma \approx 1$ is because of the high variance in the indirect estimate of the large constant $\eta^*/(1-\gamma)$. It is argued in Section V that if the error in the constants is ignored, a far tamer bound is obtained:

$$\mathsf{E}\big[\|\theta_n - \theta^*\|^2\big] \leq \frac{1}{(1-\rho^*\gamma)^p} \cdot \frac{B}{n} \tag{5}$$

A.M.D. is with the Department of Electrical Engineering, Stanford University, Stanford, CA-94305. Email: adevraj@stanford.edu

S.P.M. is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL-32611. Email: meyn@ece.ufl.edu

where $\rho^* < 1$, and $1 - \rho^*$ is an *upper bound* on the spectral gap of the transition matrix for the pair process $(\boldsymbol{X}, \boldsymbol{U})$ under the optimal policy (details are in Section V-C).

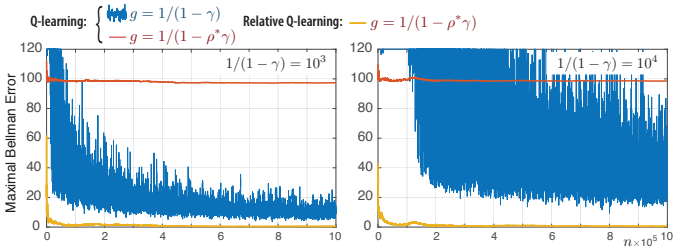

Fig. 1. Comparison of Q-learning and Relative Q-learning algorithms for the stochastic shortest path problem of [4]. The relative Q-learning algorithm is unaffected by large discounting.

The new *relative Q-learning* algorithm proposed here is designed to achieve the upper bound (5). Unfortunately, we have not yet obtained this explicit finite-n bound. We have instead obtained formulae for the *asymptotic covariance* that corresponds to each of the algorithms considered in this paper (see (9)). Numerical results in Figure 1 confirms the fast convergence of relative Q-learning in comparison to the Q-learning algorithm. More details are contained in Section VI-A.

The rest of the paper is organized as follows: The close relationship between the asymptotic covariance and sample complexity bounds is discussed in Section II-B, based on the theoretical background in Section II-A. Section III sets notation and provides background on MDPs, and Section IV contains background on Q-learning (along with new interpretations on the convergence rate of these algorithms). Section V is devoted to the new relative Q-learning algorithm. Section VI contains discussion of our results, and directions for future research and conclusions are contained in Section VII.

II. Preliminaries

A. Stochastic Approximation & Reinforcement Learning

Consider a parameterized family of $\mathbb{R}^d$-valued functions $\{\overline{f}(\theta) : \theta \in \mathbb{R}^d\}$ that can be expressed as an expectation,

$$\overline{f}(\theta) := \mathsf{E}\big[f(\theta, \Phi)\big], \qquad \theta \in \mathbb{R}^d, \tag{6}$$

with $\Phi \in \mathbb{R}^m$ a random vector, $f : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}^d$, and the expectation is with respect to the distribution of the random vector Φ. It is assumed throughout that there exists a unique vector $\theta^* \in \mathbb{R}^d$ satisfying $\overline{f}(\theta^*) = 0$. Under this assumption, the goal of SA is to estimate θ^*.

The sequence of estimates obtained from the SA algorithm are defined as follows:

$$\theta_{n+1} = \theta_n + \alpha_{n+1} f(\theta_n, \Phi_{n+1}) \tag{7}$$

where $\theta_0 \in \mathbb{R}^d$ is given, Φ_n has the same distribution as Φ for each $n \geq 0$ (or its distribution converges to that of Φ as $n \to \infty$), and $\{\alpha_n\}$ is a non-negative scalar step-size sequence. We assume $\alpha_n = g/n$ for some scalar $g > 0$, and special cases in applications to Q-learning are discussed separately in Section IV.

Asymptotic statistical theory for SA is extremely rich. Large Deviations or Central Limit Theorem (CLT) limits hold under very general assumptions for both SA and related Monte-Carlo techniques [14], [15], [16], [17], [18].

The CLT will guide design and analysis of algorithms in this paper. For a typical SA algorithm, this takes the following form: Denote the *error sequence* by

$$\tilde{\theta}_n := \theta_n - \theta^* \tag{8}$$

Under general conditions, the CLT states that the scaled sequence $\{\sqrt{n}\tilde{\theta}_n : n \geq 0\}$ converges in distribution to a multivariate Gaussian $\mathcal{N}(0, \Sigma_\theta)$. Typically, the covariance matrix of this scaled sequence is also convergent:

$$\Sigma_\theta = \lim_{n\to\infty} n\mathsf{E}[\tilde{\theta}_n \tilde{\theta}_n^\intercal] \tag{9}$$

The limit Σ_θ is known as the *asymptotic covariance*. Provided it is finite, (9) implies (2), which is the fastest possible rate [14], [15], [17], [19], [20]. For Q-learning, this also implies a bound of the form (3), but for n *"large enough"*.

An asymptotic bound such as (9) may not be satisfying for RL practitioners, given the success of finite-time performance bounds in prior research. There are however good reasons to apply this asymptotic theory in algorithm design:

(i) The asymptotic covariance Σ_θ has a simple representation as the solution to a Lyapunov equation.

(ii) The MSE convergence is refined in [21] for *linear* SA algorithms (see Section II-C): For some $\delta > 0$,

$$\Sigma_n = n^{-1}\Sigma_\theta + O(n^{-1-\delta}), \quad \text{where, } \Sigma_n := \mathsf{E}[\tilde{\theta}_n \tilde{\theta}_n^\intercal] \tag{10}$$

See [22] for general results that may be applied to many nonlinear algorithms found in RL.

(iii) The asymptotic covariance lies beneath the surface of the theory of finite-time error bounds. Here is what can be expected from the theory of large deviations [23], [24], for which the *rate function* is denoted

$$I_i(\varepsilon) := -\lim_{n\to\infty} \frac{1}{n} \log \mathsf{P}\{|\theta_n(i) - \theta^*(i)| > \varepsilon\} \tag{11}$$

The second order Taylor series approximation holds under general conditions:

$$I_i(\varepsilon) = \frac{1}{2\sigma_\theta^2(i)}\varepsilon^2 + O(\varepsilon^3) \tag{12}$$

where $\sigma_\theta^2(i) = \Sigma_\theta(i,i)$, from which we obtain

$$\begin{aligned}\mathsf{P}\{|\theta_n(i) - \theta^*(i)| > \varepsilon\} \\ = \exp\Big\{-\frac{\varepsilon^2 n}{2\sigma_\theta^2(i)} + O(n\varepsilon^3) + o(n)\Big\}\end{aligned} \tag{13}$$

where $o(n)/n \to 0$ as $n \to \infty$, and $O(n\varepsilon^3)/n$ is bounded in $n \geq 1$, and absolutely bounded by a constant times ε^3 for small $\varepsilon > 0$.

(iv) The Central Limit Theorem (CLT) holds under general assumptions:

$$\sqrt{n}\tilde{\theta}_n \xrightarrow{\text{d}} W \tag{14}$$

where the convergence is in distribution, and where W is Gaussian $\mathcal{N}(0,\Sigma_\theta)$ [15], [14]; a version of the Law of the Iterated Logarithm (LIL) also holds [25]:

$$\lim_{n\to\infty} \sqrt{\frac{n}{2\log\log n}}\tilde{\theta}_n \in C$$

where $C = \{v \in \mathbb{R}^d : v^\intercal \Sigma_\theta^{-1} v \le 1\}$. An immediate corollary is [26]:

$$\limsup_{n\to\infty} \sqrt{\frac{n}{2\log\log n}}\|\tilde{\theta}_n\| = \sqrt{\lambda_{\max}(\Sigma_\theta)} \tag{15}$$

The asymptotic theory provides insight into the slow convergence of Watkins' Q-learning algorithm, and motivates better algorithms such as Zap Q-learning [3], and the relative Q-learning algorithm introduced in Section V.

B. Sample complexity bounds

A sample complexity bound for an algorithm is defined based on the number of iterations required to obtain a desired probability of error. Consider for concreteness a single entry i of a parameter estimate in SA: for given $\delta, \varepsilon > 0$, we seek an integer $\overline{n}_i(\varepsilon,\delta)$ such that

$$\mathsf{P}\{|\theta_n(i) - \theta^*(i)| > \varepsilon\} \le \delta\,, \qquad \text{for all } n \ge \overline{n}_i(\varepsilon,\delta). \tag{16}$$

Such bounds are a foundation of statistical learning theory [27]. Below are three techniques to construct $\overline{n}$, beginning with the most common approach:

1. LDP theory The inequalities of Hoeffding and Bennett are finite-n variants of (11):

$$\mathsf{P}\{|\theta_n(i) - \theta^*(i)| > \varepsilon\} \le \bar{b}\exp(-n\bar{I}_i(\varepsilon))\,, \quad n \ge 1 \tag{17}$$

where $\bar{b}$ is a constant and $\bar{I}_i(\varepsilon) > 0$ for $\varepsilon > 0$. A sample complexity bound then follows easily, with

$$\overline{n}_i(\varepsilon,\delta) = \frac{1}{\bar{I}_i(\varepsilon)}\big[\log(\bar{b}) + \log(\delta^{-1})\big] \tag{18}$$

See for example [28], [29], [30], [31], and [32], [33] for general theory in a Markov setting.

2. MSE Given a true finite-n version of (10):

$$\mathsf{E}[(\theta_n(i) - \theta^*(i))^2] \le \overline{\sigma}^2(i) n^{-1} \tag{19}$$

A sample complexity bound follows from Chebyshev's inequality, using

$$\overline{n}_i(\varepsilon,\delta) = \frac{\overline{\sigma}^2(i)}{\varepsilon^2}\delta^{-1} \tag{20}$$

Finite-n bounds on mean-square error are contained in [6], [34], [21], [35], and the mean ℓ_∞ bound in [5] implies a similar sample complexity bound.

3. CLT A finite-n version of the CLT is the Berry-Esseen bound: for all $z > 0$,

$$\Big|\varrho_i(z,n) - 2\bar{F}(z)\Big| \le \frac{K_i}{\sqrt{n}} \tag{21}$$

where $\varrho_i(z,n)$ is the error probability with CLT scaling:

$$\varrho_i(z,n) = \mathsf{P}\big\{\sqrt{n}|\theta_n(i) - \theta^*(i)| > z\sigma_\theta(i)\big\}$$

and $\bar{F}$ is the complementary CDF for a standard Normal r.v.. For i.i.d. sequences, a simple expression for K_i is available; bounds for Markov sequences is less complete [36], [37].

For any $z > 0$ and $\delta > 2\bar{F}(z)$, denote

$$\overline{n}_i(\varepsilon,\delta,z) = \max\Big\{\frac{z^2}{\varepsilon^2}\sigma_\theta^2(i)\,,\, \frac{1}{4}\frac{K_i^2}{[\delta - 2\bar{F}(z)]^2}\Big\} \tag{22}$$

The bound (21) implies a family of sample complexity bounds that can be optimized over z: for $n \ge \overline{n}_i(\varepsilon,\delta,z)$,

$$\mathsf{P}\{|\theta_n(i) - \theta^*(i)| > \varepsilon\} \le 2\bar{F}(z) + 2\frac{K_i}{\sqrt{n}} \le \delta \tag{23}$$

The asymptotic covariance is central to each approach:

1. If the the limit (11) and the bound (17) each hold, then the rate function must dominate: $\bar{I}_i(\varepsilon) \le I_i(\varepsilon)$. To maximize this upper bound we must minimize $\sigma_\theta^2(i)$ (recall (12), and remember we are typically interested in small $\varepsilon > 0$).

2. Similarly, the mean-square error bound (19) combined with the approximation (10) implies $\overline{\sigma}^2(i) \ge \sigma_\theta^2(i)$.

3. The bound (23) requires $\sigma_\theta^2(i)$ through the definition (22).

This theory provides strong motivation for considering the asymptotic covariance Σ_θ in algorithm design.

Based on the above discussion, we conjecture that

(i) The sample-path complexity bound (18) with $\bar{I}$ quadratic is possible for Watkins' algorithm, provided we use $\alpha_n = [1 + (1-\gamma)n]^{-1}$ in the right hand side of the update equation (1). This step-size was proposed independently in [5], [38].

(ii) With relative Q-learning, we can obtain similar sample complexity result with $\alpha_n = [1 + (1-\rho^*)n]^{-1}$, which is *independent of* γ.

However, for more complex algorithms we do not expect to obtain tight bounds, with $\bar{I}_i(\varepsilon) \approx I_i(\varepsilon)$. For this reason we advocate the CLT for algorithm design and evaluation, even without a sharp Berry-Esseen bound. We frequently find that the CLT is highly predictive of parameter error, where the covariance $\sigma_\theta^2(i)$ is estimated via independent runs. Fig. 2 shows results from one experiment using the relative Q-learning algorithm: the histograms were obtained based on 10^3 independent runs, with time horizons ranging from $N = 10^3$ to 10^6. The CLT approximation is good even for the shortest run, and nearly perfect for $N \ge 10^4$.

C. Explicit Mean Square Error bounds for SA

We first present a special case of the main result of [21] for linear SA algorithms, and then an extension to nonlinear SA. These results are later recalled in applications to Q-learning: Even in the tabular setting (see Section IV-A for definitions), the Q-learning algorithm is a *nonlinear* SA algorithm, due to a minimization that appears in the recursion.

The analysis of the SA recursion (7) begins with the transformation to (1), with $\Delta_{n+1} = f(\theta_n, \Phi_{n+1}) - \overline{f}(\theta_n)$. The difference $f(\theta\,, \Phi_{n+1}) - \overline{f}(\theta)$ has zero mean for any (deterministic) $\theta \in \mathbb{R}^d$ when Φ_{n+1} has the same distribution as Φ (recall (6)). Though the results of [21] extend to Markovian noise, for the purposes of this paper, we assume that $\{\Delta_n\}$ is a martingale difference sequence:

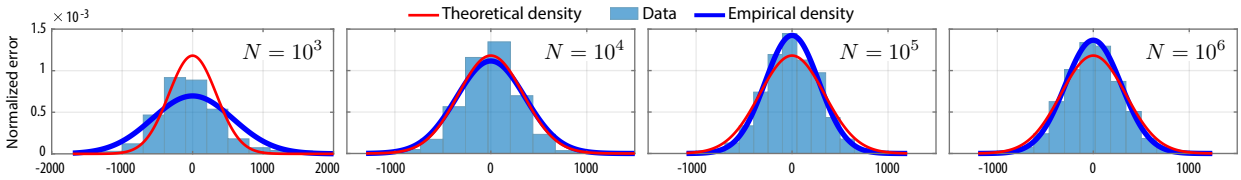

Fig. 2. Histogram of $\{\sqrt{N}\tilde{\theta}_N(i)\}$ for 10^3 independent runs. The CLT approximation is good even for the shortest run, and nearly perfect for $N \geq 10^4$.

(A1) The sequence $\{\Delta_n : n \geq 1\}$ is a martingale difference sequence. Moreover, for some $\bar{\sigma}_\Delta^2 < \infty$ and any initial condition $\theta_0 \in \mathbb{R}^d$,

$$\mathsf{E}[\|\Delta_{n+1}\|^2 \mid \Delta_1, \ldots, \Delta_n] \leq \bar{\sigma}_\Delta^2(1 + \|\theta_n\|^2), \qquad n \geq 0$$

We also assume a scalar, diminishing step-size sequence:

(A2) $\alpha_n = g/n$, for some scalar $g > 0$, and all $n \geq 1$

With Σ_n defined in (10), denote

$$\sigma_n^2 = \text{trace}\,(\Sigma_n) = \mathsf{E}[\|\tilde{\theta}_n\|^2]$$

We say $\sigma_n^2 \to 0$ at rate $1/n^\mu$ (with $\mu > 0$), if for each $\varepsilon > 0$,

$$\lim_{n\to\infty} n^{\mu-\varepsilon}\sigma_n^2 = 0 \qquad \textit{and} \qquad \lim_{n\to\infty} n^{\mu+\varepsilon}\sigma_n^2 = \infty \tag{24}$$

It is known that the maximal value is $\mu = 1$.

The analysis in [21] is based on a "linearized" approximation of the SA recursion (7):

$$\theta_{n+1} = \theta_n + \alpha_{n+1}\big[A_{n+1}\theta_n - b_{n+1}\big] \tag{25}$$

where, $A_{n+1} = \mathcal{A}(\Phi_{n+1})$ is a $d \times d$ matrix, and $b_{n+1} = b(\Phi_{n+1})$ is $d \times 1$. Let A and b denote the respective means:

$$A = \mathsf{E}[\mathcal{A}(\Phi)]\,, \qquad b = \mathsf{E}[b(\Phi)] \tag{26}$$

where the expectations are in steady state. We assume that the $d \times d$ matrix A is Hurwitz, a necessary condition for convergence of (25):

(A3) The $d \times d$ matrix A is Hurwitz. □

(A3) implies that A is invertible, and $\theta^* = A^{-1}b$.

The recursion (25) can be rewritten in the form (1):

$$\theta_{n+1} = \theta_n + \alpha_{n+1}\big[A\theta_n - b + \Delta_{n+1}\big] \tag{27}$$

in which $\{\Delta_n\}$ is the noise sequence:

$$\Delta_{n+1} = A_{n+1}\theta^* - b_{n+1} + \tilde{A}_{n+1}\tilde{\theta}_n \tag{28}$$

with $\tilde{A}_{n+1} = A_{n+1} - A$. The parameter error sequence also evolves as a simple linear recursion:

$$\tilde{\theta}_{n+1} = \tilde{\theta}_n + \alpha_{n+1}[A\tilde{\theta}_n + \Delta_{n+1}] \tag{29}$$

The asymptotic covariance (9) exists under special conditions (see Thm. II.1), and under these conditions it satisfies the Lyapunov equation

$$(gA + \tfrac{1}{2}I)\Sigma_\theta + \Sigma_\theta(gA + \tfrac{1}{2}I)^\intercal + g^2\Sigma_\Delta = 0 \tag{30}$$

where the "noise covariance matrix" Σ_Δ is defined to be

$$\Sigma_\Delta = \mathsf{E}\big[\big(A_{n+1}\theta^* - b_{n+1}\big)\big(A_{n+1}\theta^* - b_{n+1}\big)^\intercal\big] \tag{31}$$

Thm. II.1 is a special case of the main result of [21] (which does not impose the martingale assumption (A1)).

Theorem II.1. *Suppose (A1) – (A3) hold. Then the following hold for the linear recursion (29), for each initial $(\Phi_0, \tilde{\theta}_0)$:*

(i) *If Real$(\lambda) < -\frac{1}{2}$ for every eigenvalue λ of gA, then*

$$\Sigma_n = n^{-1}\Sigma_\theta + O(n^{-1-\delta})$$

where $\delta = \delta(A, \Sigma_\Delta) > 0$, and $\Sigma_\theta \geq 0$ is the solution to the Lyapunov equation (30). Consequently, $\mathsf{E}[\|\tilde{\theta}_n\|^2]$ converges to zero at rate $1/n$.

(ii) *Suppose there is an eigenvalue λ of gA that satisfies*

$$-\varrho_0 = \textit{Real}(\lambda) > -\tfrac{1}{2}$$

Let $\nu \neq 0$ denote a corresponding left eigenvector, and suppose that $\Sigma_\Delta\nu \neq 0$. Then, $\mathsf{E}[|\nu^\intercal\tilde{\theta}_n|^2]$ converges to 0 at a rate $1/n^{2\varrho_0}$. Consequently, $\mathsf{E}[\|\tilde{\theta}_n\|^2]$ converges to zero at rate no faster than $1/n^{2\varrho_0}$. □

Prop. II.2 extends the conclusions of Thm. II.1 to nonlinear SA (1). The proof is contained in Appendix A.

Proposition II.2. *Consider the general SA algorithm (1). Suppose (A1) – (A3) hold with $A := \partial_\theta \overline{f}(\theta)|_{\theta=\theta^*}$, and that $\overline{f}$ has the form*

$$\overline{f}(\theta) = -\theta + \bar{F}(\theta)\,, \qquad \theta \in \mathbb{R}^d$$

with $\bar{F}$ Lipschitz continuous, a strict contraction, and C^1 in a neighborhood of the origin. Then,

(i) If Real$(\lambda) < -\frac{1}{2}$ for every eigenvalue λ of gA, then

(a) The CLT holds for $\{W_n = \sqrt{n}\tilde{\theta}_n\}$, with asymptotic covariance $\Sigma_\theta \geq 0$ the solution to the Lyapunov equation (30).

(b) Weak convergence goes beyond bounded and continuous functions: for any measurable function $g\colon \mathbb{R}^d \to \mathbb{R}$ with at most quadratic growth we have

$$\lim_{n\to\infty} \mathsf{E}[g(W_n)] = \mathsf{E}[g(W_\infty)]\,, \qquad W_\infty \sim N(0, \Sigma_\theta)$$

In particular, $\mathsf{E}[\|\tilde{\theta}_n\|^2]$ converges to zero at rate $1/n$, and

$$\lim_{n\to\infty} n\Sigma_n = \Sigma_\theta$$

(ii) Suppose there is an eigenvalue λ of gA that satisfies

$$-\varrho_0 = \textit{Real}(\lambda) > -\tfrac{1}{2}$$

Let $\nu \neq 0$ denote a corresponding left eigenvector, and suppose that $\Sigma_\Delta\nu \neq 0$. Then, $\mathsf{E}[|\nu^\intercal\tilde{\theta}_n|^2]$ converges to 0 at a rate $1/n^{2\varrho_0}$. Consequently, $\mathsf{E}[\|\tilde{\theta}_n\|^2]$ converges to zero at rate no faster than $1/n^{2\varrho_0}$.

It seems likely that the finite-n bound in Thm. II.1 also holds for the nonlinear SA algorithm. We believe that coupling

techniques of [7] is one way to establish such results for Q-learning. More importantly, even though the finite-n result remains a conjecture, we have already highlighted how the CLT is often predictive of finite-n performance.

III. Markov Decision Processes Formulation

Consider a Markov Decision Processes (MDP) model with state space X, action space U, cost function $c\colon \mathsf{X}\times\mathsf{U}\to\mathbb{R}$, and discount factor $\gamma\in(0,1)$. It is assumed throughout that the state and action spaces are finite: denote $\ell=|\mathsf{X}|$ and $\ell_u=|\mathsf{U}|$. In the following, the terms 'action', 'control', and 'input' are used interchangeably.

Along with the state-action process $(\boldsymbol{X},\boldsymbol{U})$ is an i.i.d. sequence $\boldsymbol{I}=\{I_1,I_2,\dots\}$ used to model a randomized policy. It is assumed without loss of generality that each I_n takes values in a finite set. An input sequence $\boldsymbol{U}$ is called *non-anticipative* if

$$U_n=z_n(X_0,U_0,I_1\dots,U_{n-1},X_n,I_n)\,,\qquad n\ge 0$$

where $\{z_n\}$ is a sequence of functions.

Under the assumption that the state and action spaces are finite, it follows that there are a finite number of deterministic stationary policies: $\{\phi^{(i)}:1\le i\le\ell_\phi\}$, where each $\phi^{(i)}:\mathsf{X}\to\mathsf{U}$, and $\ell_\phi\le(\ell_u)^\ell$. A randomized stationary policy is defined by a probability mass function (pmf) μ on the $\{1,2,\dots,\ell_\phi\}\times\mathsf{X}$, such that

$$U_n=\sum_{k=1}^{\ell_\phi}\iota_n(k)\phi^{(k)}(X_n)\tag{32}$$

with $\mu(k,x)=\mathsf{P}\{\iota_n(k)=1\mid X_0\,,\dots\,,X_{n-1},X_n=x\}$ for each $n\ge0$, $1\le k\le\ell_\phi$, and $x\in\mathsf{X}$. It is assumed that ι_n is a fixed function of (I_n,X_n) for each n.

For each $u\in\mathsf{U}$, the controlled transition matrix P_u acts on functions $V\colon\mathsf{X}\to\mathbb{R}$ via

$$\begin{aligned}P_uV\,(x)&:=\sum_{x'\in\mathsf{X}}P_u(x,x')V(x')\\&=\mathsf{E}[V(X_{n+1})\,|\,X_n=x\,,U_n=u\,;X_k,I_k,U_k\colon k<n]\end{aligned}$$

where the second equality holds for any non-anticipative input sequence $\boldsymbol{U}$. For any deterministic stationary policy ϕ, let S_ϕ denote the substitution operator, defined for any function $q\colon\mathsf{X}\times\mathsf{U}\to\mathbb{R}$ by

$$S_\phi q\,(x):=q(x,\phi(x))$$

If the policy ϕ is randomized, of the form (32), we then define

$$S_\phi q\,(x)=\sum_k\mu(k)q(x,\phi^{(k)}(x))$$

With P viewed as a single matrix with $\ell\cdot\ell_u$ rows and ℓ columns, and S_ϕ viewed as a matrix with ℓ rows and $\ell\cdot\ell_u$ columns, the following interpretations hold:

Lemma III.1. *Suppose that $\boldsymbol{U}$ is defined using a stationary policy ϕ (possibly randomized). Then, both $\boldsymbol{X}$ and the pair process $(\boldsymbol{X},\boldsymbol{U})$ are Markovian, and*

(i) *$P_\phi:=S_\phi P$ is the transition matrix for $\boldsymbol{X}$.*
(ii) *PS_ϕ is the transition matrix for $(\boldsymbol{X},\boldsymbol{U})$.* □

A. Q-function and the Bellman Equation

For any (possibly randomized) stationary policy ϕ, we consider two value functions

$$V_\phi(x):=\sum_{n=0}^{\infty}(\gamma P_\phi)^nS_\phi c\,(x)\tag{33a}$$

$$Q_\phi(x,u):=\sum_{n=0}^{\infty}(\gamma PS_\phi)^nc\,(x,u)\tag{33b}$$

which are related via

$$Q_\phi(x,u)=c(x,u)+\gamma P_uV_\phi\,(x)\tag{34}$$

The function $V_\phi:\mathsf{X}\to\mathbb{R}$ in (33a) is the value function that corresponds to the policy ϕ (with the corresponding transition probability matrix P_ϕ), and cost function $S_\phi c$, that appears in TD-learning algorithms [1], [39]. The function $Q_\phi:\mathsf{X}\times\mathsf{U}\to\mathbb{R}$ is the fixed-policy Q-function considered in the SARSA algorithm [40], [41], [42].

The minimal (optimal) value function $V^*(x):=\min_\phi V_\phi(x)$ is the unique solution to the Bellman equation:

$$V^*(x)=\min_u\Big\{c(x,u)+\gamma\sum_{x'\in\mathsf{X}}P_u(x,x')V^*(x')\Big\}\tag{35}$$

Any minimizer defines a deterministic stationary policy $\phi^*\colon\mathsf{X}\to\mathsf{U}$ that is optimal over all input sequences [13]:

$$\phi^*(x)\in\arg\min_u\Big\{c(x,u)+\gamma\sum_{x'\in\mathsf{X}}P_u(x,x')V^*(x')\Big\}\tag{36}$$

The Q-function associated with V^* is given by (34) with $\phi=\phi^*$, which is precisely the term within the brackets in (35):

$$Q^*(x,u):=c(x,u)+\gamma P_uV^*\,(x)$$

The Bellman equation (35) implies a similar fixed point equation for the Q-function:

$$Q^*(x,u)=c(x,u)+\gamma P_u\underline{Q}^*(x)\tag{37}$$

in which $\underline{Q}(x):=\min_u Q(x,u)$ for any $Q\colon\mathsf{X}\times\mathsf{U}\to\mathbb{R}$.

For any function $q\colon\mathsf{X}\times\mathsf{U}\to\mathbb{R}$, let $\phi^q\colon\mathsf{X}\to\mathsf{U}$ denote an associated policy that satisfies

$$\phi^q(x)\in\arg\min_u q(x,u)\tag{38}$$

It is assumed to be specified *uniquely* as follows:

$$\begin{aligned}&\phi^q:=\phi^{(\kappa)}\text{ such that}\\&\kappa=\min\{i:\phi^{(i)}(x)\in\arg\min_u q(x,u),\text{ for all }x\in\mathsf{X}\}\end{aligned}\tag{39}$$

Using the above notations, and the definitions in Lemma III.1, the fixed point equation (37) can be rewritten as

$$Q^*(x,u)=c+\gamma PS_{\phi^*}Q^*(x,u),\ \text{ where }\phi^*=\phi^q,\ q=Q^*\tag{40}$$

IV. Q-LEARNING

The goal in Q-learning is to approximately solve the fixed point equation (37), *without* assuming knowledge of the controlled transition matrix. We restrict the discussion here to the case of linear parameterization for the Q-function: $Q^\theta(x,u) = \theta^\intercal \psi(x,u)$, where $\theta \in \mathbb{R}^d$ denotes the parameter vector, and $\psi\colon \mathsf{X} \times \mathsf{U} \to \mathbb{R}^d$ denotes the vector of basis functions.

For a given parameter vector $\theta \in \mathbb{R}^d$, let $\mathcal{B}^\theta : \mathsf{X} \times \mathsf{U} \to \mathbb{R}$ denote the corresponding Bellman error:

$$\mathcal{B}^\theta(x,u) := c(x,u) + \gamma P_u \underline{Q}^\theta(x) - Q^\theta(x,u) \tag{41}$$

A *Galerkin approach* to approximating the optimal Q-function Q^* is formulated as follows: Obtain a non-anticipative input sequence $\boldsymbol{U}$ (using a randomized stationary policy ϕ), and a d-dimensional stationary stochastic process $\boldsymbol{\zeta}$ that is adapted to $(\boldsymbol{X}, \boldsymbol{U})$. The *Galerkin relaxation* of the fixed point equation (37) is the root finding problem: Find θ^* such that,

$$\overline{f}_i(\theta^*) := \mathsf{E}\Big[\widetilde{\mathcal{B}}^{\theta^*}_{n+1} \zeta_n(i)\Big] = 0\,, \qquad 1 \le i \le d \tag{42}$$

where, for each $\theta \in \mathbb{R}^d$, $\widetilde{\mathcal{B}}^\theta_{n+1}$ is the "temporal difference"

$$\widetilde{\mathcal{B}}^\theta_{n+1} := c(X_n, U_n) + \gamma \underline{Q}^\theta(X_{n+1}) - Q^\theta(X_n, U_n)\,, \tag{43}$$

and the expectation in (42) is with respect to the steady state distribution of $(\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{\zeta})$. Equation (42) is often called the *projected Bellman equation*. It is a special case of the general root-finding problem that is the focus of SA algorithms.

The following Q(0) algorithm is the SA algorithm (7), applied to estimate θ^* that solves (42): For initialization $\theta_0 \in \mathbb{R}^d$, define the sequence of estimates recursively:

$$\theta_{n+1} = \theta_n + \alpha_{n+1} \zeta_n \widetilde{\mathcal{B}}^{\theta_n}_{n+1}\,, \qquad \zeta_n = \psi(X_n, U_n) \tag{44}$$

The choice for the sequence of eligibility vectors $\{\zeta_n\}$ in (44) is inspired by the TD(0) algorithm [43], [1].

For a sequence of $d \times d$ matrices $\boldsymbol{G} = \{G_n\}$, the *matrix-gain* Q(0) *algorithm* is described as follows: For initialization $\theta_0 \in \mathbb{R}^d$, the sequence of estimates are defined recursively:

$$\theta_{n+1} = \theta_n + \alpha_{n+1} G_{n+1} \psi(X_n, U_n) \widetilde{\mathcal{B}}^{\theta_n}_{n+1} \tag{45}$$

A common choice is

$$G_n = \left(\frac{1}{n} \sum_{k=1}^{n} \psi(X_k, U_k) \psi^\intercal(X_k, U_k) \right)^{-1} \tag{46}$$

The success of these algorithms has been demonstrated in a few restricted settings, such as optimal stopping [44], [45], [46], deterministic optimal control [47], and the tabular setting that is imposed throughout the remainder of the paper.

We assume a tabular setting throughout this work.

A. Tabular Q-learning

The basic Q-learning algorithm of Watkins [48], [49] (also known as *"tabular" Q-learning*) is a particular instance of the Galerkin approach (44). The basis functions are taken to be indicator functions:

$$\psi_i(x,u) = \mathbb{I}\{(x,u) = (x^i, u^i)\}\,, \quad 1 \le i \le d \tag{47}$$

where $\{(x^k, u^k) : 1 \le k \le d\}$ is an enumeration of all state-input pairs, with $d = \ell \cdot \ell_u$. The goal of this approach is to *exactly* compute the function Q^*. Letting ϖ denote the invariant pmf of the Markov chain $(\boldsymbol{X}, \boldsymbol{U})$, and substituting $\zeta_n \equiv \psi(X_n, U_n)$ with ψ defined in (47), the objective (42) can be rewritten as follows: Find $\theta^* \in \mathbb{R}^d$ such that, for each $1 \le i \le d$,

$$0 = \mathsf{E}\big[\widetilde{\mathcal{B}}^{\theta^*}_{n+1} \psi_i(X_n, U_n)\big] \tag{48}$$

$$= \Big[c(x^i, u^i) + \gamma \mathsf{E}\big[\underline{Q}^{\theta^*}(X_{n+1}) | X_n = x^i\,, U_n = u^i\big] - Q^{\theta^*}(x^i, u^i) \Big] \varpi(x^i, u^i) \tag{49}$$

where the expectation in (48) is in steady state. The conditional expectation in (49) is

$$\mathsf{E}\big[\underline{Q}^{\theta^*}(X_{n+1}) | X_n = x^i\,, U_n = u^i\big] = P_{u^i} \underline{Q}^{\theta^*}(x^i)$$

Consequently, (49) can be rewritten as

$$0 = \mathcal{B}^{\theta^*}(x^i, u^i) \varpi(x^i, u^i) \tag{50}$$

If $\varpi(x^i, u^i) > 0$ for each $1 \le i \le d$, then the function Q^{θ^*} that solves (50) is identical to the optimal Q-function in (37).

There are three flavors of Watkins' Q-learning that are common in the literature. We discuss each of them below.

Asynchronous Q-learning: The SA algorithm applied to solve (48) coincides with the most basic version of Watkins' Q-learning algorithm: For initialization $\theta_0 \in \mathbb{R}^d$, define the sequence of estimates $\{\theta_n : n \ge 0\}$ recursively:

$$\theta_{n+1} = \theta_n + \alpha_{n+1} \widetilde{\mathcal{B}}^{\theta_n}_{n+1} \psi(X_n, U_n) \tag{51}$$

Algorithm (51) coincides with the Q(0) algorithm (44), with ψ defined in (47). Based on this choice of basis functions, a single entry of θ is updated at each iteration, corresponding to the state-input pair (X_n, U_n) observed (hence the term "asynchronous"). The parameter θ can be identified with the function Q^θ in this tabular setting. This equivalence justifies a slight abuse of notation: replace Q^θ by Q and set $\widetilde{\mathcal{B}}^Q_{n+1} = \widetilde{\mathcal{B}}^\theta_{n+1}$ (defined in (43)), resulting in a more familiar form of (51):

$$Q^{n+1}(X_n, U_n) = Q^n(X_n, U_n) + \alpha_{n+1} \widetilde{\mathcal{B}}^{Q^n}_{n+1} \tag{52}$$

and $Q^{n+1}(x,u) = Q^n(x,u)$ if $(x,u) \ne (X_n, U_n)$.

With $\alpha_n = 1/n$, the *ODE approximation* of (51) takes the form (see [2] for details):

$$\tfrac{d}{dt} q_t(x,u) = \varpi(x,u) \Big[c(x,u) + \gamma P_u \underline{q}_t(x) - q_t(x,u) \Big] \tag{53}$$

in which $\underline{q}_t(x) = \min_u q_t(x,u)$ as defined below (37). We recall in Section IV-B conditions under which this ODE is stable, and explain why we cannot expect a finite asymptotic covariance in typical settings.

A second and perhaps more popular "Q-learning flavor" is defined using a particular "state-action dependent" step-size [7], [31], [38]. For each (x,u), denote $\alpha_n(x,u) = 0$ if the pair (x,u) has not been visited up until time $n-1$. Otherwise,

$$\alpha_n(x,u) = \frac{1}{n(x,u)}\,, \quad n(x,u) = \sum_{j=0}^{n-1} \mathbb{I}\{X_j = x, U_j = u\} \tag{54}$$

The ODE approximation of (52) simplifies with (54):

$$\tfrac{d}{dt}q_t(x,u) = c(x,u) + \gamma P_u \underline{q}_t\,(x) - q_t(x,u) \tag{55}$$

The asynchronous variant of Watkins' Q-learning algorithm (51) with step-size (54) can be viewed as an instance of G-Q(0) algorithm defined in (45), with the matrix gain sequence (46), and step-size $\alpha_n = 1/n$. On substituting the Watkins' basis defined in (47), we find that this matrix is diagonal: $G_n = \widehat{\Pi}_n^{-1}$, where

$$\widehat{\Pi}_n(i,\,i) = \frac{1}{n}\sum_{k=1}^{n} \mathbb{I}\{X_k = x^i, U_k = u^i\}\,, \qquad 1 \le i \le d$$

By the Law of Large Numbers, we have

$$\lim_{n\to\infty} G_n = \lim_{n\to\infty} \widehat{\Pi}_n^{-1} = \Pi^{-1} \tag{56}$$

where Π is a diagonal matrix with entries $\Pi(i,i) = \varpi(x^i,\,u^i)$. It is easy to see why the ODE approximation (53) simplifies to (55) with this matrix gain.

Synchronous Q-learning: In this final flavor, each entry of the Q-function approximation is updated in each iteration. It is popular in the literature because the analysis is greatly simplified.

The algorithm requires a model that provides the next state of the Markov chain, conditioned on any given current state-action pair: let $\{X_n^i : n \ge 1, 1 \le i \le d\}$ denote a collection of mutually independent random variables taking values in X. Assume moreover that for each i, the sequence $\{X_n^i : n \ge 1\}$ is i.i.d. with common distribution $P_{u^i}(x^i, \cdot\,)$. The *synchronous Q-learning* algorithm is then obtained as follows: For initialization $\theta_0 \in \mathbb{R}^d$, define the sequence of estimates $\{\theta_n : n \ge 0\}$ recursively:

$$\begin{aligned}\theta_{n+1} = \theta_n + \alpha_{n+1}\sum_{i=1}^{d}\big[c(x^i,u^i) + \gamma \underline{Q}^{\theta_n}(X_{n+1}^i) \\ - Q^{\theta_n}(x^i,u^i)\big]\psi(x^i\,,u^i)\end{aligned} \tag{57}$$

Once again, based on the choice of basis functions (47), and observing that θ is identified with the estimate Q^θ, an equivalent form of the update rule (57) is

$$\begin{aligned}Q^{n+1}(x^i,u^i) = Q^n(x^i,u^i) + \alpha_{n+1}\big[c(x^i,u^i) \\ + \gamma \underline{Q}^n(X_{n+1}^i) - Q^n(x^i,u^i)\big],\quad 1 \le i \le d\end{aligned} \tag{58}$$

Using the step-size $\alpha_n = 1/n$ we obtain the simple ODE approximation (55).

B. Convergence and Rate of Convergence

Convergence of the tabular Q-learning algorithms can be established under the following assumptions:

(Q1) The input $\boldsymbol{U}$ is defined by a randomized stationary policy of the form (32). The joint process $(\boldsymbol{X}, \boldsymbol{U})$ is an irreducible Markov chain. That is, it has a unique invariant pmf ϖ satisfying $\varpi(x,u) > 0$ for each x, u.

(Q2) The optimal policy ϕ^* is unique. □

Both ODEs (53) and (55) are stable under assumption (Q1) [50], which then (based on the results of [2]) implies that $\boldsymbol{\theta}$ converges to Q^* a.s.. To obtain rates of convergence requires an examination of the linearization of the ODEs at their equilibrium.

Linearization is justified under Assumption (Q2), which implies the existence of $\varepsilon > 0$ such that

$$\phi^*(x) = \arg\min_{u\in\mathsf{U}} Q^\theta(x,u)\,, \quad \text{if} \quad \|Q^\theta - Q^*\| < \varepsilon \tag{59}$$

Lemma IV.1. *Under Assumptions (Q1) and (Q2) the following approximations hold*

(i) *When $\|q_t - Q^*\| < \varepsilon$, the ODE (53) reduces to*

$$\tfrac{d}{dt}q_t = -\Pi[I - \gamma P S_{\phi^*}]q_t - b$$

where Π is defined below (56), and $b(x,u) = -\varpi(x,u)c(x,u)$, expressed as a $d\times 1$ column vector.

(ii) *When $\|q_t - Q^*\| < \varepsilon$, the ODE (55) reduces to*

$$\tfrac{d}{dt}q_t = -[I - \gamma P S_{\phi^*}]q_t - b$$

where $b(x,u) = -c(x,u)$.

The proof is contained in Appendix B.

The definition of the linearization matrix A in (26) is extended to non-linear functions as follows [4], [51]:

$$A = \partial_\theta \overline{f}(\theta)\big|_{\theta=\theta^*}$$

The crucial take-away from Lemma IV.1 is the linearization matrix that corresponds to each tabular Q-learning algorithms:

$$A = -\Pi[I - \gamma P S_{\phi^*}] \quad \text{in case (i) of Lemma IV.1} \tag{60a}$$

$$A = -[I - \gamma P S_{\phi^*}] \quad \text{in case (ii) of Lemma IV.1} \tag{60b}$$

Since $\gamma < 1$, and PS_{ϕ^*} is a transition matrix of an irreducible Markov chain (see Lemma III.1), it follows that both matrices are Hurwitz.

We consider next conditions under which the asymptotic covariance for Q-learning is *not* finite. The noise covariance matrix Σ_Δ defined in (31) is diagonal in all three flavors of Q-learning discussed in Section IV-A. For the asynchronous Q-learning algorithm (52) with step-size (54), or the synchronous Q-learning algorithm (58), the diagonal elements of Σ_Δ are given by $\Sigma_\Delta^{s(i,i)} =$

$$\begin{aligned}&\gamma^2\mathsf{E}\Big[\big(\underline{Q}^*(X_{n+1}) - P_{u^i}\underline{Q}^*(x_i)\big)^2\Big|X_n = x^i, U_n = u^i\Big] \\ =&\gamma^2\mathsf{E}\Big[\big(V^*(X_{n+1}) - P_{u^i}V^*(x_i)\big)^2\Big|(X_n = x^i, U_n = u^i)\Big]\end{aligned} \tag{61}$$

The noise covariance for asynchronous Q-learning with step-size $\alpha_n = 1/n$ is $\Sigma_\Delta^a = \Pi\Sigma_\Delta^s\Pi$, with Π defined below (56).

Theorem IV.2. *Suppose that assumptions (Q1) and (Q2) hold, and $\alpha_n \equiv 1/n$. Then, the sequence of parameters $\{\theta_n\}$ obtained using the asynchronous Q-learning algorithm (51) converges to Q^* a.s.. Suppose moreover that the conditional variance of $V^*(X_n)$ is positive:*

$$\sum_{x,x',u} \varpi(x,u)P_u(x,x')[V^*(x') - P_uV^*\,(x)]^2 > 0 \tag{62}$$

and

$$(1-\gamma)\max_{x,u}\varpi(x,u) < \tfrac{1}{2} \tag{63}$$

Then,

(i) *The asymptotic covariance of the algorithm is infinite:*

$$\lim_{n\to\infty} n\mathsf{E}[\|\theta_n - \theta^*\|^2] = \infty$$

(ii) $\mathsf{E}[\|\theta_n - \theta^*\|^2]$ *converges to zero at a rate no faster than* $1/n^{2(1-\gamma)}$. ☐

The inequality (63) is satisfied for $\gamma \geq \frac{1}{2}$.

Thm. IV.2 explains why the Q-learning algorithm can be terribly slow: If the discount factor is close to 1, which is typical in many applications, using a step-size of the form $\alpha_n = 1/n$ results in a MSE convergence rate that is *much* slower than the optimal rate $1/n$.

Similar conclusions hold for the other flavors of tabular Q-learning, for which the algorithm admits the ODE approximation (55). Based on Lemma IV.1, the linearization matrix for these algorithms is defined in (60b). This poses problems when $\gamma > \frac{1}{2}$, but for these algorithms there is a simple remedy:

Theorem IV.3. *For asynchronous Q-learning with the step-size rule (54), or synchronous Q-learning with step-size $\alpha_n = 1/n$, the matrix shown in (60b) is equal to the linearization matrix $A = \partial_\theta \overline{f}(\theta)\big|_{\theta=\theta^*}$. It has one eigenvalue $\lambda_1 = -(1-\gamma)$, and $\mathrm{Re}(\lambda(A)) < -(1-\gamma)$ for every other eigenvalue. Consequently,*

(i) *Subject to (62), the asymptotic covariance is not finite whenever $\gamma > \frac{1}{2}$.*

(ii) *Suppose that the step-sizes are scaled: use $\alpha_n(x,u) = [(1-\gamma)n(x,u)]^{-1}$ for asynchronous Q-learning, or $\alpha_n = [(1-\gamma)n]^{-1}$ for synchronous Q-learning. Then, the eigenvalue test passes: for each eigenvalue $\lambda = \lambda(A)$,*

$$\mathrm{Re}(\lambda) = -(1-\gamma)^{-1}\mathrm{Re}\big(\lambda([I - \gamma P S_{\phi^*}])\big) \leq -1$$

The resulting asymptotic covariance is obtained as a solution to the Lyapunov equation (30), with $g = (1-\gamma)^{-1}$, and $\Sigma_\Delta = \Sigma^{\mathrm{s}}_\Delta$ defined in (61). ☐

The step-size rule $\alpha_n = [(1-\gamma)n]^{-1}$ is equivalent to $\alpha_n = [1 + (1-\gamma)n]^{-1}$ that appears in [5], in the sense that each algorithm will share the same asymptotic covariance.

Overview of proofs: We begin with Thm. IV.2. The proof of convergence can be found in [48], [52], [2]. The proof of infinite asymptotic covariance is based on an application of Prop. II.2. A brief overview follows.

To establish the slow convergence rate, an eigenvector for A (defined in (60a)) can be constructed with strictly positive entries, and with real part of the corresponding eigenvalue satisfying $\mathrm{Re}(\lambda) \geq -1/2$ (see Appendix A.2 of [4]). Interpreted as a function $v\colon \mathsf{X}\times\mathsf{U} \to \mathbb{C}$, this eigenvector satisfies $v^\dagger \Sigma_\Delta v =$

$$\gamma^2 \sum_{x,u,x'} \varpi(x,u)|v(x,u)|^2 P_u(x,x')[V^*(x') - P_u V^*(x)]^2 \quad (64)$$

where Σ_Δ is the noise covariance matrix (recall (61)), and $v^\dagger$ denotes complex-conjugate transpose. Assumption (62) ensures that the right hand side of (64) is strictly positive, as required in part (ii) of Prop. II.2.

Thm. IV.3 is based on the simple structure of the eigenvalues of the linearization matrix $A = -[I - \gamma P S_{\phi^*}]$ defined in (60b). Because PS_{ϕ^*} is the transition matrix for an irreducible Markov chain, it follows that all of its eigenvalues are in the closed unit disk in the complex plane, with a single eigenvalue at $\lambda = 1$. Consequently, A has a single eigenvalue at $\lambda = -(1-\gamma)$, and $\mathrm{Re}(\lambda(A)) < -(1-\gamma)$ for all other eigenvalues. An application of Prop. II.2 then implies both (i) and (ii) of the theorem. ☐

Theorems IV.2 and IV.3 motivate the introduction of new algorithms whose performance does not degrade with large γ.

V. Relative Q-learning

The following *relative Bellman equation* was inspired by the decomposition (4):

$$H^*(x,u) = c(x,u) + \gamma P_u \underline{H}^*(x) - \delta\langle \mu\,, H^*\rangle \quad (65)$$

where $\delta > 0$ is a positive scalar, $\mu : \mathsf{X}\times\mathsf{U} \to [0,1]$ is a pmf (both design choices), and $\langle \mu\,, H^*\rangle = \sum_{x,u} \mu(x,u) H^*(x,u)$. For example, we may choose μ concentrated at some fixed pair $(x^\bullet, u^\bullet)$, so that $\langle \mu, H\rangle = H(x^\bullet, u^\bullet)$ for any H.

With $\gamma = 1$, the fixed point equation (65) is very similar to the fixed point equation that appears in the average cost Q-learning formulation of [53], though the motivations are different: the prior work is devoted to Q-learning algorithm for the average cost criterion, while the present paper concerns reliable algorithms in the discounted cost setting. Motivation for the relative Q-function is similar to the introduction of normalization to define the *advantage function* of RL [54].

Define $\widetilde{H}^*(x,u) := Q^*(x,u) - \langle \mu\,, Q^*\rangle$, which by (4) can be expressed

$$\widetilde{H}^*(x,u) = \widetilde{Q}^*(x,u) - \langle \mu\,, \widetilde{Q}^*\rangle$$

It follows that $\widetilde{H}^*$ is uniformly bounded in γ, x, and u [11], [13]. The relationship (i) in Prop. V.1 is immediate from the definitions. Part (ii) implies that H^* is uniformly bounded over $\gamma \in [0,1)$. Observe that (66) implies that Q^* can be recovered from H^* and μ.

Proposition V.1. *Under (Q1)–(Q2), the solution H^* to (65) is unique, and satisfies:*

(i) $H^*(x,u) = Q^*(x,u) - k$, *with*

$$k = \frac{\delta}{1+\delta-\gamma}\langle \mu\,, Q^*\rangle = \frac{\delta}{1-\gamma}\langle \mu\,, H^*\rangle \quad (66)$$

(ii) $H^*(x,u) = \widetilde{H}^*(x,u) + \eta^*/\delta + o(1)$, *where $o(1) \to 0$ as $\gamma \uparrow 1$.*

☐

Proof. The proof of (i) follows from (65) and (37). This further implies

$$\begin{aligned} H^*(x,u) &= Q^*(x,u) - \Big(1 - \frac{1-\gamma}{1+\delta-\gamma}\Big)\langle \mu\,, Q^*\rangle \\ &= \widetilde{H}^*(x,u) + \frac{1}{1+\delta-\gamma}(1-\gamma)\langle \mu\,, Q^*\rangle \end{aligned}$$

This concludes the proof of (ii), since $(1-\gamma)\langle \mu\,, Q^*\rangle \to \eta^*$ as $\gamma \uparrow 1$; this well known fact follows from (4) (see also [11]). □

The objective in relative Q-learning is to estimate H^*. Since Q^* and H^* differ only by a constant, the policy ϕ^* defined in (40) satisfies $\phi^* = \phi^q$, with $q = H^*$ (see (38)). It is therefore irrelevant whether we estimate Q^* or H^*, if we are ultimately interested only in the optimal policy.

We conjecture that estimating H^* results in finite-n error bounds of the form (5), which is uniformly bounded for all $\gamma < 1$ (in sharp contrast to finite-n bounds for estimating Q^*—recall (3)). We establish here that the asymptotic covariance is uniformly bounded in γ under the right choices for δ and the step-size.

A. Relative Q-learning Algorithm

Consider a linear parameterization for the relative Q-function: $H^\theta(x,u) = \theta^\intercal \psi(x,u)$, where $\theta \in \mathbb{R}^d$ denotes the parameter vector, and $\psi\colon \mathsf{X}\times\mathsf{U} \to \mathbb{R}^d$ denotes the vector of basis functions. We restrict the discussion here to the tabular case, where the basis functions $\{\psi_i : 1 \le i \le d\}$ are the indicator functions defined in (47).

The goal in *tabular relative Q-learning* is to find θ^* such that

$$\overline{f}(\theta^*) := \mathsf{E}\big[\{c(X_n,U_n) + \gamma \underline{H}^{\theta^*}(X_{n+1}) - \delta\langle \mu\,, H^{\theta^*}\rangle - H^{\theta^*}(X_n,U_n)\}\psi(X_n,U_n)\big] = 0 \tag{67}$$

where $\boldsymbol{U}$ is a non-anticipative input sequence (obtained using a randomized stationary policy ϕ), $\underline{H}^\theta(x) = \min_u H^\theta(x,u)$, and the expectation is with respect to the steady state distribution of the Markov chain $(\boldsymbol{X},\boldsymbol{U})$. With the basis functions chosen to be indicator functions (47), interpretations similar to (48)–(50) hold, and the objective (67) can be rewritten as: For each $1 \le i \le d$,

$$\overline{f}_i(\theta^*) = \Big[c(x^i,u^i) + \gamma P_{u^i}\underline{H}^{\theta^*}(x^i) - \delta\langle \mu, H^{\theta^*}\rangle - H^{\theta^*}(x^i,u^i)\Big]\varpi(x^i,u^i) = 0 \tag{68}$$

where ϖ denotes the invariant pmf of $(\boldsymbol{X},\boldsymbol{U})$.

We once again assume (Q1) and (Q2) of Section IV-B throughout. Under (Q1), it is easy to see that H^{θ^*} that solves (68) is identical to the optimal relative Q-function in (65). Assumption (Q2) implies existence of $\varepsilon > 0$ such that

$$\phi^*(x) = \arg\min_{u\in\mathsf{U}} H^\theta(x,u)\,, \qquad \|H^\theta - H^*\| < \varepsilon \tag{69}$$

As in Section IV-A, there are many flavors of relative Q-learning algorithm that are possible. We restrict our discussion here to the *asynchronous relative Q-learning* algorithm, which requires access to a single sample path of the Markov chain $(\boldsymbol{X},\boldsymbol{U})$. Extension of the results and discussion to other flavors of the algorithm is straightforward.

Asynchronous Relative Q-learning: The asynchronous algorithm is a direct application of SA to solve (67): For initialization $\theta_0 \in \mathbb{R}^d$, define the sequence of estimates $\{\theta_n : n \ge 0\}$ recursively:

$$\theta_{n+1} = \theta_n + \alpha_{n+1}\Big[c(X_n,U_n) + \gamma \underline{H}^{\theta_n}(X_{n+1}) - \delta\langle \mu\,, H^{\theta_n}\rangle - H^{\theta_n}(X_n,U_n)\Big]\psi(X_n,U_n) \tag{70}$$

Based on the choice of basis functions (47), a single entry of θ is updated at each iteration, corresponding to the state-input pair (X_n,U_n) observed. By identifying θ with the estimate H^θ, we can rewrite (70) as

$$H^{n+1}(X_n,U_n) = H^n(X_n,U_n) + \alpha_{n+1}\big[c(X_n,U_n) + \gamma \underline{H}^n(X_{n+1}) - \delta\langle \mu, H^n\rangle - H^n(X_n,U_n)\big] \tag{71}$$

With $\alpha_n = 1/n$, the *ODE approximation* of (71) takes the form

$$\tfrac{d}{dt}h_t(x,u) = \varpi(x,u)\Big[c(x,u) + \gamma P_u \underline{h}_t\,(x) - \delta\langle \mu\,, h_t\rangle - h_t(x,u)\Big] \tag{72}$$

in which $\underline{h}_t(x) = \min_u h_t(x,u)$. Based on the discussion in Section IV-A, a "more efficient" relative Q-learning flavor is defined using a particular state-action dependent step-size (54). The ODE approximation (72) simplifies in this case:

$$\tfrac{d}{dt}h_t(x,u) = c(x,u) + \gamma P_u \underline{h}_t\,(x) - \delta\langle \mu\,, h_t\rangle - h_t(x,u) \tag{73}$$

Henceforth we restrict discussion to the relative Q-learning algorithm with a scaling of this specific step-size: $\alpha_n(x,u) = g\cdot\big[n(x,u)\big]^{-1}$ with $g > 0$. We initially assume $g = 1$.

B. Stability and Convergence of Relative Q-learning

Convergence of the algorithm holds under mild conditions:

Theorem V.2 (Stability & Convergence)**.** *Consider the relative Q-learning algorithm (71) with step-size $\alpha_n(x,u)$ satisfying (54). Then, $\lim_{n\to\infty} H^n = H^*$, a.s., for each initial condition.* □

The proof of the theorem follows from [2, Theorems 2.1 and 2.2], which tells us that stability of the ODE (73) implies firstly that

$$\sup_n \sup_{x,u} H^n(x,u) < \infty \qquad a.s.$$

and then convergence follows from more well known arguments. Global asymptotic stability of the ODE is established in Prop. A.1. A martingale noise assumption is imposed on the SA recursions considered in [2], [17] (it is argued that the stability result holds for more general Markovian noise). This extension is not required to prove Thm. V.2, as we can cast the relative Q-learning algorithm precisely within the setting of [2].

The algorithm in (71), with step-size rule (54) can be rewritten as:

$$H^{n+1}(x,u) = H^n(x,u) + \alpha_{n+1}(x,u)[\overline{f}(H^n,X_n,U_n;x,u) + \Delta_{n+1}(x\,,u)] \tag{74}$$

where

$$\overline{f}_{H^n}(X_n,U_n;x,u) = \big[\widetilde{T}H^n(x,u) - H^n(x,u)\big]\mathbb{I}\{X_n = x, U_n = u\}$$

for any H,

$$\widetilde{T}H(x,u) := c(x,u) + \gamma P_u \underline{H}(x) - \delta\langle \mu, H\rangle \tag{75}$$

and $\{\Delta_n\}$ is the noise sequence: $\Delta_{n+1}(x,u) =$

$$\gamma\Big(\underline{H}^n(X_{n+1}) - P_u\underline{H}^n(x)\Big)\mathbb{I}\{X_n = x, U_n = u\} \tag{76}$$

The recursion (74) is SA with Markovian noise.

For the purpose of analysis, it is best to visualize the algorithm (74) with step-size rule (54) as "d parallel stochastic approximation algorithms", one for each state-action pair (x,u). If a particular $(X_n\,, U_n)$ is observed in the n^{th} iteration, then the corresponding H-value is updated, with the rest of the H-values left unchanged.

The martingale difference property is expressed as follows:

$$\mathsf{E}[\Delta_{n+1}(x,u)|\mathcal{F}_n] = 0\,, \qquad \textit{for each } (x,u) \in \mathsf{X}\times\mathsf{U}, \tag{77}$$

where $\mathcal{F}_n = \sigma(X_m, U_m : m \le n)$. A second assumption of [2] also holds: for some constant $K > 0$,

$$\mathsf{E}[\|\Delta_{n+1}(x,u)\|^2|\mathcal{F}_n] \le K(1 + \|H^n\|^2) \tag{78}$$

C. Convergence Rate of Relative Q-learning

We now analyze the asymptotic covariance of the relative Q-learning algorithm (71) that approximates the ODE (73). Following along the lines of analysis in Section IV-B, the covariance analysis requires two ingredients: identification of the noise covariance Σ_Δ in (31), and examination of the linearization of the ODE (73). Recall that a finite asymptotic covariance depends on properties of the eigenvalues of the linearization matrix $A = \partial_\theta \overline{f}(\theta)\big|_{\theta=\theta^*}$.

As for the first ingredient, it follows from (76) that the noise covariance is a diagonal matrix, with $\Sigma_\Delta^{(i,i)} =$

$$\gamma^2\mathsf{E}\Big[\big(\underline{H}^*(X_{n+1}) - P_{u^i}\underline{H}^*(x^i)\big)^2 \mid (X_n, U_n) = (x^i, u^i)\Big] \tag{79}$$

This is identical to the noise covariance in Watkins' algorithm:

Lemma V.3. *The noise covariance matrix Σ_Δ^q for the Q-learning algorithm (defined in (61)), and Σ_Δ^h for the relative Q-learning algorithm (defined in (79)) are identical.*

Proof. The proof is a direct application of Prop. V.1: with $\kappa_\gamma = \delta\langle \mu\,, H^*\rangle/(1-\gamma)$ we obtain, for each $1 \le i \le d$,

$$\begin{aligned}
\Sigma_\Delta^{q\ (i,i)} &= \gamma^2\mathsf{E}\Big[\big(\underline{Q}^*(X_{n+1}) - P_{u^i}\underline{Q}^*(x^i)\big)^2 \,|\, X_n = x^i, U_n = u^i\Big] \\
&= \gamma^2\mathsf{E}\Big[\big(\underline{H}^*(X_{n+1}) + \kappa_\gamma - P_{u^i}\underline{H}^*(x^i)\big)^2 - \kappa_\gamma \,|\, X_n = x^i, U_n = u^i\Big] \\
&= \Sigma_\Delta^{h\ (i,i)}
\end{aligned}$$

□

We henceforth denote $\Sigma_\Delta = \Sigma_\Delta^q = \Sigma_\Delta^h$.

We turn next to the linearization of the ODE (73) at its equilibrium: this is justified under Assumption (Q2), which implies the existence of $\varepsilon > 0$ such that (69) holds. The following result is a direct analog of Lemma IV.1 for the relative Q-learning algorithm.

Lemma V.4. *Under Assumption (Q2), when $\|\tilde{h}_t\| < \varepsilon$, with $\varepsilon > 0$ used in (69), the ODE (73) simplifies to*

$$\tfrac{d}{dt}h_t = -[I - \gamma PS_{\phi^*} + \delta\cdot \mathbb{1}\otimes\mu]h_t - b$$

where $b(x,u) = -c(x,u)$. □

In Lemma V.4, $\mathbb{1} \in \mathbb{R}^d$ is viewed as a column vector with each component $\mathbb{1}_i = 1$, $1 \le i \le d$, and $\otimes$ denotes the outer product. The lemma provides a simple expression for the linearization matrix:

$$A = -[I - \gamma PS_{\phi^*} + \delta\cdot \mathbb{1}\otimes\mu] \tag{80}$$

In addition to (Q1) and (Q2), we impose the following additional assumption for the convergence rate analysis:

(Q3) The Markov chain with transition matrix PS_{ϕ^*} is uni-chain: the eigenspace corresponding to the eigenvalue $\lambda_1 = 1$ is one-dimensional. □

Denote

$$\rho^* = \max\{\mathrm{Re}(\lambda_i) : i \ge 2\} \tag{81}$$

where the maximum is over all eigenvalues of PS_{ϕ^*} except $\lambda_1 = 1$. Under (Q3) we have $\rho^* < 1$, and in fact $\rho^* < 0$ is possible. Let ρ denote the magnitude of the second largest eigenvalue of PS_{ϕ^*}:

$$\rho = \max\{|\lambda_i| : \lambda_i \ne 1\} \tag{82}$$

The scalar ρ is also known as the *mixing rate* of the Markov chain $(\boldsymbol{X}\,, \boldsymbol{U})$, with the input sequence $\boldsymbol{U}$ defined by ϕ^*, and $1-\rho$ is the *spectral gap* of the corresponding transition matrix. While $\rho^* < 1$ is always true under (Q3), this does not exclude the possibility that $\rho = 1$ (i.e., there is no spectral gap). We have an obvious bound:

Lemma V.5. *The quantities ρ and ρ^* defined in (81) and (82) satisfy $\rho^* \le \rho$.*

The bound is achieved if there is a real and positive eigenvalue satisfying $\lambda_2 = \rho$.

The following theorem (which is analogous to Thm. IV.3 for the Q-learning algorithm) is the main result of this subsection.

Theorem V.6. *For the asynchronous relative Q-learning algorithm (71) with step-size rule (54), the matrix A in (80) is equal to the linearization matrix $A = \partial_\theta \overline{f}(\theta)\big|_{\theta=\theta^*}$. If we choose $\delta \ge \gamma(1-\rho^*)$, then each eigenvalue of A satisfies $\mathrm{Re}(\lambda(A)) \le -(1-\gamma\rho^*)$. Consequently,*

(i) *The asymptotic covariance is infinite if $\gamma\rho^* > \frac{1}{2}$, and also $\nu_2^\dagger\Sigma_\Delta\nu_2 > 0$, where ν_2 is an eigenvector of PS_{ϕ^*} with eigenvalue satisfying $Re(\lambda_2) = \rho^*$.*

(ii) *Suppose that the step-sizes are scaled:*

$$\alpha_n(x,u) = [(1-\gamma\rho^*)\cdot n(x,u)]^{-1} \tag{83}$$

Then, the eigenvalue test passes: each eigenvalue $\lambda(A)$ satisfies

$$\begin{aligned}
\mathrm{Re}(\lambda(A)) &= -(1-\gamma\rho^*)^{-1}\mathrm{Re}\big(\lambda\big([I - \gamma PS_{\phi^*} - \delta\cdot\mathbb{1}\otimes\mu]\big)\big) \\
&\le -1
\end{aligned}$$

The asymptotic covariance of the resulting algorithm is obtained as a solution to the Lyapunov equation (30), with $g = (1-\gamma\rho^*)^{-1}$, and Σ_Δ defined in (79). □

To be clear: *the condition $\rho < 1$ is not necessary for stability of relative Q-learning, or uniform boundedness of*

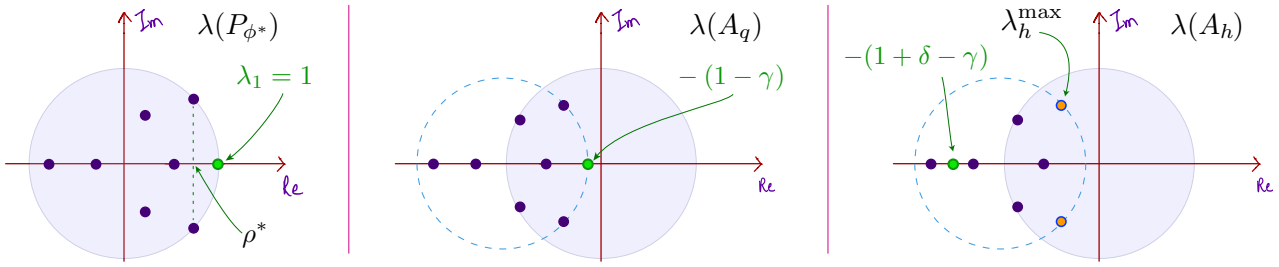

Fig. 3. Relationship between the eigenvalues of the matrices PS_{ϕ^*}, A_q, and A.

the asymptotic covariance. Consider the example illustrated in Fig. 3. The plot of eigenvalues for PS_{ϕ^*} shown on the left hand side indicates complex eigenvalues on the unit circle, so that $\rho = 1$. The plots show that $\rho^* < 1$, and therefore, $-(1-\gamma\rho^*) < -(1-\gamma)$. In this case, Thm. V.6 (ii) implies that the relative Q-learning algorithm with step-size $\alpha_n = g \cdot [n(x,u)]^{-1}$, $g = -[1-\gamma\rho^*]^{-1}$ will have finite asymptotic covariance.

We close the section with proof of Thm. V.6.

Proof of Thm. V.6: The proof is based on comparing the eigenvalues of the matrix A with the eigenvalues of the linearization matrix that corresponds to the asynchronous Watkins' Q-learning algorithm (recall Lemma IV.1 (ii), and Eq. (60b)):

$$A_q = -[I - \gamma P S_{\phi^*}] \tag{84}$$

Lemma V.7. (i) *The matrix A_q is Hurwitz. Furthermore, there exists a single eigenvalue at $\lambda = -(1-\gamma)$, and all other eigenvalues satisfy*

$$\mathrm{Re}(\lambda(A_q)) \le -(1-\gamma\rho^*) \tag{85}$$

where $\rho^* \in [0,1)$ is defined in (81).

(ii) *The vector $\mathbb{1}$ is a right eigenvector of A, with eigenvalue $\lambda_1 = -(1-\gamma+\delta)$. Moreover, every eigenvalue λ of A, that is* not *equal to $-(1-\gamma+\delta)$, is also an eigenvalue of A_q, with identical left eigenvectors.*

Proof. The proof of (i) follows from the following observations: Thm. IV.3 combined with assumption (Q3) establishes the upper bound $\mathrm{Re}(\lambda(A_q)) \le -(1-\gamma)$. The column vector $\mathbb{1}$ is an eigenvector, whose eigenvalue coincides with this identity: $A_q\mathbb{1} = -(1-\gamma)\cdot\mathbb{1}$.

We now prove (ii). The first claim follows from these steps:

$$A\mathbb{1} = -[I - \gamma P S_{\phi^*} + \delta\cdot\mathbb{1}\otimes\mu]\mathbb{1} = -(1-\gamma+\delta)\cdot\mathbb{1}$$

If $\lambda \ne -(1-\gamma+\delta)$ is an eigenvalue of A_q, with corresponding left eigenvector ν, we have:

$$\begin{aligned}\lambda\nu^\intercal = \nu^\intercal A &= -\nu^\intercal[I - \gamma P S_{\phi^*} - \delta\cdot\mathbb{1}\otimes\mu] \\ &\overset{(a)}{=} -\nu^\intercal[I - \gamma P S_{\phi^*}] \\ &= \nu^\intercal A_q\end{aligned}$$

where (a) follows from orthogonality of $\mathbb{1}$ and the left eigenvector ν. ∎

Lemma V.8 asserts that (85) holds for *every eigenvalue* in the relative Q-learning algorithm if δ is greater than or equal to $1-\rho^*$. The choice $\delta = \gamma$ will always satisfy the condition in Lemma V.8. The proof is immediate from Lemma V.7.

Lemma V.8. *Suppose we choose $\delta \ge \gamma(1-\rho^*)$. Then, each eigenvalue of the linearization matrix A defined in (80) satisfies*

$$\mathrm{Re}(\lambda(A)) \le -(1-\gamma\rho^*) \tag{86}$$

Consequently, the matrix A is Hurwitz, for all $0 < \gamma < 1/\rho^*$. ∎

Proof of Thm. V.6. Lemma V.8 proves the first conclusion in Thm. V.6: Each eigenvalue of the linearization matrix A of relative Q-learning satisfies $\mathrm{Re}(\lambda(A)) \le -(1-\gamma\rho^*)$. The proof of (i) and (ii) then follow from Prop. II.2. ∎

VI. Discussion

Theorems IV.3 and V.6 contain conditions for finite asymptotic covariance of the Q-learning and relative Q-learning algorithms. Here we provide a more quantitative comparison. We begin with a coarse comparison, considering the trace of the respective covariance matrices.

Proposition VI.1. *Denote by $\Sigma_\theta^q(g)$, $\Sigma_\theta^h(g)$, the asymptotic covariance matrices for Q-learning and relative Q-learning with $\alpha_n = g\cdot[n(x,u)]^{-1}$. Each is finite for all sufficiently large g, and satisfy the following bounds, uniformly in γ:*

$$\min_g \left\{\mathrm{trace}\left(\Sigma_\theta^q(g)\right)\right\} \ge O\left(\frac{\mathrm{trace}\,(\Sigma_\Delta^2)}{(1-\gamma)^2}\right), \quad \textit{sub. to (62)} \tag{87a}$$

$$\min_g \left\{\mathrm{trace}\left(\Sigma_\theta^h(g)\right)\right\} \le O\left(\frac{\mathrm{trace}\,(\Sigma_\Delta^2)}{(1-\rho^*\gamma)^2}\right) \tag{87b}$$

Proof. The proof of eqn. (87a) follows from Prop. VI.2. This result also implies that the lower bound for $\Sigma_\theta^q(g)$ in (87a) is attained with $g_q = (1-\gamma)^{-1}$. The upper bound (87b) is a simple consequence of Thm. V.6. ∎

These bounds show a significant contrast in performance when $1/(1-\gamma) \gg 1/(1-\rho^*)$. However, we find that the two covariance matrices actually coincide on a subspace. This is made precise in the following subsections.

A. Covariance Comparison on a Single Eigenspace

The stark contrast between the two covariance matrices is made clear here. Denote by λ_{h_1} the eigenvalue of the matrix $A_h = A$ (defined in (80)) that has the largest real part (marked

with pink circles in the third part of Fig. 3), and ν_{h_1} the corresponding left-eigenvector. Similarly, denote λ_{q_1} to be the eigenvalue of A_q (defined in (84)) that has the largest real part (the green circle in the second part of Fig. 3), and ν_{q_1} the corresponding left-eigenvector. Our interest here is the magnitude of the non-negative quantities

$$\sigma_q^2(1,1) := \nu_{q_1}^\dagger \Sigma_\theta^q \nu_{q_1} \qquad \textit{and} \qquad \sigma_h^2(1,1) := \nu_{h_1}^\dagger \Sigma_\theta^h \nu_{h_1} \quad (88)$$

Explicit formulae are easily obtained, and then optimized over g. Analogous formulae are obtained in Section VI-C for other eigenvectors, so we omit the proof of (89) here.

Proposition VI.2.

$$\sigma_q^2(1,1) = g^2 \frac{\sigma_{\Delta_q}^2(1,1)}{1-2g(1-\gamma)}\,, \quad g > [2(1-\gamma)]^{-1} \quad (89a)$$

$$\sigma_h^2(1,1) = g^2 \frac{\sigma_{\Delta_h}^2(1,1)}{1-2g(1-\gamma\rho^*)}\,, \quad g > [2(1-\gamma\rho^*)]^{-1} \quad (89b)$$

where $\sigma_{\Delta_q}^2(1,1) := \nu_{q_1}^\dagger \Sigma_\Delta \nu_{q_1}$ *and* $\sigma_{\Delta_h}^2(1,1) := \nu_{h_1}^\dagger \Sigma_\Delta \nu_{h_1}$. *The minimizing gains are given by* $g_q = (1-\gamma)^{-1}$ *in* (89a)*, and* $g_h = (1-\gamma\rho^*)^{-1}$ *in* (89b)*. This results in the minimal values,*

$$\min_g \sigma_q^2(1,1) = \frac{\sigma_{\Delta_q}^2(1,1)}{(1-\gamma)^2}\,, \quad \min_g \sigma_h^2(1,1) = \frac{\sigma_{\Delta_h}^2(1,1)}{(1-\rho^*\gamma)^2} \quad (90)$$

□

The gains introduced in Prop. VI.2 imply $\mathrm{Re}(\lambda) \le -1$ for any eigenvalue of either $g_q A_q$ or $g_h A_h$. Denote by $(\lambda_{h\mathbb{1}}, \nu_{h\mathbb{1}})$ the eigenvalue/left-eigenvector pair of the matrix A_h, with corresponding right-eigenvector $\mathbb{1}$. Similarly, denote by $(\lambda_{q\mathbb{1}}, \nu_{q\mathbb{1}})$ the eigenvalue/left-eigenvector pair of the matrix A_q, with corresponding right-eigenvector $\mathbb{1}$. These eigenvalues correspond to the green circles in Figure 3, and $(\lambda_{q\mathbb{1}}, \nu_{q\mathbb{1}})$ coincides with $(\lambda_{q_1}, \nu_{q_1})$. However, a similar property *does not* hold for the matrix A_h. We compare next the variances on the eigenspace spanned by $\mathbb{1}$:

$$\sigma_{q\mathbb{1}}^2 := \nu_{q\mathbb{1}}^\dagger \Sigma_\theta^q \nu_{q\mathbb{1}} \qquad \textit{and} \qquad \sigma_{h\mathbb{1}}^2 := \nu_{h\mathbb{1}}^\dagger \Sigma_\theta^h \nu_{h\mathbb{1}} \quad (91)$$

Proposition VI.3. *Consider the Q-learning and relative Q-learning with step-size scaling* g_q *and* g_h *defined in Prop. VI.2, and with* $\delta \ge \gamma(1-\rho^*)$ *in* (67)*. Then,*

$$\sigma_{q\mathbb{1}}^2 = \frac{\sigma_{\Delta_{q\mathbb{1}}}^2}{(1-\gamma)^2} \qquad \sigma_{h\mathbb{1}}^2 = \frac{\sigma_{\Delta_h\mathbb{1}}^2}{(1-\rho^*\gamma)\big(1+\rho^*\gamma-2(\gamma-\delta)\big)} \quad (92)$$

where, $\sigma_{\Delta_{q\mathbb{1}}}^2 = \sigma_{\Delta_q}^2(1,1) = \nu_{q\mathbb{1}}^\dagger \Sigma_\Delta \nu_{q\mathbb{1}}$, $\sigma_{\Delta_h\mathbb{1}}^2 = \nu_{h\mathbb{1}}^\dagger \Sigma_\Delta \nu_{h\mathbb{1}}$. □

For the choice of $\delta \ge \gamma(1-\rho^*)$, we have $\sigma_{h\mathbb{1}}^2 \le \sigma_h^2(1,1)$ defined in (89b), consistent with (87b) of Prop. VI.1.

In Fig. 1 we compare the performance of Q-learning and relative Q-learning algorithms applied to a simple 6-state MDP that was considered in [3, Section 3]. Experiments were run for $\gamma = 0.999$ and $\gamma = 0.9999$, and in each of the two cases, we implemented Q-learning with optimized step-size $\alpha_n = g_q/n$, $g_q = 1/(1-\gamma)$, and relative Q-learning with optimized step-size $\alpha_n = g_h/n$, $g_h = 1/(1-\rho^*\gamma)$. In addition, we also implemented Q-learning with $\alpha_n = g_h/n$; the motivation is discussed in the following subsection.

We return now to Figure 2, which shows histograms of $\{\sqrt{N}\tilde{\theta}_N(i)\}$ from the relative Q-learning algorithm (obtained from 10^3 independent runs with random initial conditions, up to time horizon 10^6, with data collected at this value, and intermediate values $N = 10^3, 10^4, 10^5$). The theoretical pdf's were obtained based on the CLT (14): the distribution of $\sqrt{N}\tilde{\theta}_N$ is approximated by $\mathcal{N}(0, \Sigma_\theta)$ for large N, with Σ_θ obtained as a solution to (30). The figure makes clear that the CLT predicts finite-N behavior for N as small as 10^4. This is remarkable, but not surprising given the prior work [3], [38].

B. Solidarity on a Subspace

Prop. VI.2 again shows that a larger gain g is required in Watkins' algorithm, and we can expect a larger asymptotic covariance. Prop. VI.3 compares the asymptotic covariance with optimal gains for the two algorithms on a particular subspace. The question we ask here is: *what about the remainder of* $\mathbb{R}^d$*?*

The asymptotic covariances appearing in Prop. VI.1 solve the respective Lyapunov equations:

$$0 = F_q \Sigma_\theta^q + \Sigma_\theta^q F_q^\intercal + g^2 \Sigma_\Delta \quad (93a)$$

$$0 = F_h \Sigma_\theta^h + \Sigma_\theta^h F_h^\intercal + g^2 \Sigma_\Delta \quad (93b)$$

where $F_q = gA_q + \frac{1}{2}I$ and $F_h = gA_h + \frac{1}{2}I$. It is shown in Prop. VI.4 that the solutions are identical on the subspace $\mathbb{R}_0^d = \{v \in \mathbb{R}^d : v^\dagger \mathbb{1} = 0\}$ in the sense that

$$v^\dagger \Sigma_\theta^q w = v^\dagger \Sigma_\theta^h w\,, \qquad \textit{for all } v, w \in \mathbb{R}_0^d \quad (94)$$

This identity is valid even when F_q *is not Hurwitz*, so that Σ_θ^q is not finite valued. The proof makes use of the representations

$$\begin{aligned} v^\dagger \Sigma_\theta^q w &= g^2 \int_0^\infty v^\dagger e^{F_q t} \Sigma_\Delta e^{F_q^\intercal t} w \, dt \\ v^\dagger \Sigma_\theta^h w &= g^2 \int_0^\infty v^\dagger e^{F_h t} \Sigma_\Delta e^{F_h^\intercal t} w \, dt \end{aligned} \quad (95)$$

Proposition VI.4. *Suppose that the matrix* F_h *is Hurwitz. Then the asymptotic covariance* Σ_θ^h *exists and is finite, and moreover* (94) *holds, subject to the definition of* (95)*.*

Proof. Given the representation (95), it is enough to establish

$$v^\dagger e^{tF_q} = v^\dagger e^{tF_h} \qquad \textit{for all } t > 0 \textit{ and } v \in \mathbb{R}_0^d \quad (96)$$

The proof makes use of the following identity:

$$v^\dagger F_q = v^\dagger F_h\,, \qquad \textit{for all } v \in \mathbb{R}_0^d \quad (97)$$

Moreover, $F_q^\dagger \colon \mathbb{R}_0^d \to \mathbb{R}_0^d$ and $F_h^\dagger \colon \mathbb{R}_0^d \to \mathbb{R}_0^d$.

These identities imply many others. Starting from $v^\dagger F_q = v^\dagger F_h$ for $v \in \mathbb{R}_0^d$, we obtain $v^\dagger F_q F_h = v^\dagger F_h^2$, and the identity $v^\dagger F_q^2 = v^\dagger F_h^2$ follows since $(v^\dagger F_q)^\dagger \in \mathbb{R}_0^d$. By induction we obtain $v^\dagger F_q^n = v^\dagger F_h^n$ for each n and each $v \in \mathbb{R}_0^d$, and then (96) follows from the Taylor series representation of the matrix exponential. □

Prop. VI.4 says that in the subspace that is orthogonal to the column vector $\mathbb{1}$, the solutions to the Lyapunov equations in (93b) are identical, *provided we use the same scalar gain*.

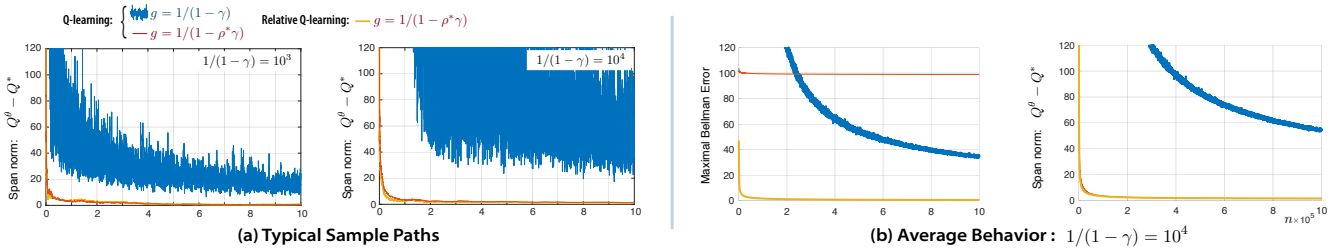

Fig. 4. (a) Span norm error for Q-learning and Relative Q-learning are similar, even for $\gamma \sim 1$. (b) Average error for the three algorithms, with $1/(1-\gamma) = 10^4$.

Along with Prop. VI.3, this implies that the challenge faced when estimating the Q-function, as opposed to the relative Q-function is due to estimating a constant. If an algorithm designer is content with ignoring the error in constants, they can obtain a significantly lower asymptotic covariance Q-learning algorithm.

Fig. 4 (a) contains plots of the span-semi-norm of the errors in the Q-function estimates obtained using Q-learning and relative Q-learning algorithms. Once again, experiments were run for $\gamma = 0.999$ and $\gamma = 0.9999$, and for each γ, we used two different step-sizes for the Q-learning algorithm: $g_q = 1/(1-\gamma)$, and $g_h = 1/(1-\rho^*\gamma)$. Since the span semi-norm ignores the error in the constants, we notice that the performance of the Q-learning and relative Q-learning algorithms with the same step-size $g_h = 1/(1-\rho^*\gamma)$ is very similar — consistent with our findings in Prop. VI.4. Figures 1 and 4 (a) illustrate the behavior of each of the three algorithms on a single sample path. Fig. 4 (b) shows the average error of obtained from $N = 10^3$ independent runs of each algorithm.

C. What if the Transition Matrix is Diagonalizable?

If the matrix PS^*_ϕ is diagonalizable, this means that there is a basis consisting of eigenvectors, and also a basis consisting of left-eigenvectors. Viewed as column vectors, we find that $d-1$ of the left eigenvectors span $\mathbb{R}^d_0$. From this we obtain a refinement of Prop. VI.4: a solution to the Lyapunov equation on $\mathbb{R}^d_0$, and on all of $\mathbb{R}^d$ when F_q is Hurwitz.

If PS^*_ϕ is diagonalizable, then the definition (84) implies that the same is true for A_q. Let $\{\nu_i : 1 \le i \le d\}$ be a basis of left eigenvectors for A_q, with corresponding eigenvalues $\{\lambda_i : 1 \le i \le d\}$, and suppose the eigenvalues are ordered so that $\lambda_1(A_q) = -(1-\gamma)$.

Lemma V.7 (ii) asserts that $\{\nu_i : 2 \le i \le d\}$ are also left eigenvectors for A_h, with common left eigenvalues. Moreover, $\nu_i^\dagger \mathbb{1} = 0$ for $2 \le i \le d$, so that their span equals $\mathbb{R}^d_0$.

For each $2 \le i, j \le d$, consider the quantities:

$$\sigma_q^2(i,j) := \nu_i^\dagger \Sigma_\theta^q \nu_j \qquad \sigma_h^2(i,j) := \nu_i^\dagger \Sigma_\theta^h \nu_j$$
$$\sigma_\Delta^2(i,j) := \nu_i^\dagger \Sigma_\Delta \nu_j \tag{98}$$

The identity $\sigma_q^2(i,j) = \sigma_h^2(i,j)$ follows from Prop. VI.4. Multiplying the left hand side of (93a) and (93b) by $\nu_i^\dagger$, and the right hand side by ν_j, we obtain

$$\sigma_q^2(i,j) = \sigma_h^2(i,j) = g^2 \frac{\sigma_\Delta^2(i,j)}{1 - g(\lambda_i + \lambda_j)} \tag{99}$$

For the optimal gains g_q and g_h appearing in Prop. VI.2, substitution into (99) gives the approximation when $\gamma \approx 1$: For $2 \le i, j \le d$,

$$\sigma_q^2(i,j) = O\left(\frac{\sigma_\Delta^2(i,j)}{1-\gamma}\right), \quad \sigma_h^2(i,j) = O\left(\frac{\sigma_\Delta^2(i,j)}{1-\rho^*\gamma}\right) \tag{100}$$

D. Performance Comparison via Total Discounted Returns

So far, Figures 1 and 4 illustrate convergence rate of the estimates of the Q-functions. As noted earlier, even though the limits of the two algorithms are different, the policies induced by the optimal Q-function and the relative Q-function are identical.

Shown in Fig. 5 are plots of the average total discounted rewards obtained with the Q-function estimates; the x-axis indicates the number of iterations (log scale). The plot was obtained by implementing $N = 100$ parallel simulations of the Q-learning and relative Q-learning algorithms with their optimized step-sizes g_q and g_h, and in each sample path, after $n = 10^k$ iterations, $k = 1, \dots, 6$, the average total discounted reward $\eta_\gamma(\phi)$ that corresponds to the policy ϕ induced by the Q-function estimates was computed using Monte-Carlo:

$$\eta_\gamma(\phi) := \mathsf{E}\left[\sum_{t=0}^{\infty} \gamma^t R(X_t, \phi(X_t))\right] \tag{101}$$

where the expectation was estimated using $M = 10$ roll-outs and computing the empirical average, and the infinite sum within the expectation was replaced with a finite sum until $T = 10^4$ for $\gamma = 0.999$ and $T = 5 \times 10^4$ for $\gamma = 0.9999$.

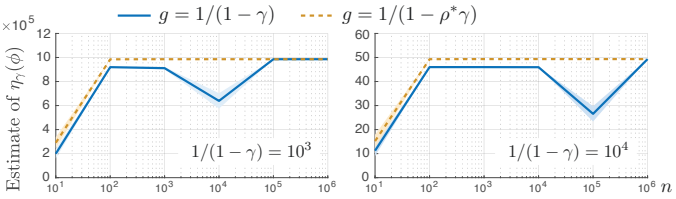

Fig. 5. Estimates of $\eta_\gamma(\phi)$ defined in (101) with $\phi = \phi^q$ and $\phi = \phi^h$, the policies that are induced by the Q-function and relative Q-function estimates with $g = 1/(1-\gamma)$ and $g = 1/(1-\rho^*\gamma)$ respectively.

The reliability of relative Q-learning is remarkable: estimates reach the optimal in about $n = 100$ iterations, and remain there for all larger n. The sequence of policy estimates obtained from Q-learning also reach an *almost-optimal* one quickly, but consistently take a large excursion resulting in very poor performance, and finally returning to the optimal policy after 10^5 iterations ($\gamma = 0.999$) and 10^6 iterations ($\gamma = 0.9999$).

VII. Conclusions and Future Work

The factor $1/(1-\gamma)^p$ is ubiquitous in RL complexity bounds, where $p \geq 2$. We have shown that this dependency is *artificial*: if we ignore the constant terms (that does not affect the optimal policy), this factor can be improved to $1/(1-\rho^*\gamma)^p$, where $\rho^* < 1$ under very general conditions. Specifically, we showed that the classical Q-learning algorithm of Watkins has asymptotic (CLT) variance that grows as a quadratic in $1/(1-\gamma)$, and the relative Q-learning algorithm has asymptotic variance that is bounded by a quadratic in $1/(1-\rho^*\gamma)$. We believe that this will lead to comparable improvements in sample complexity bounds.

The techniques introduced in this work can also be extended to various other RL algorithms. For example, it is straightforward to modify the recursion (70) to obtain a *relative TD(0)-learning* algorithm for a discounted cost MDP, with linear function-approximation. The choice of μ may require care in a continuous state-space setting.

The contributions of this paper are complementary to the Zap-Q techniques of [3]. In view of the matrix gain Q-learning algorithm (45), the goal in [3] is to obtain an optimal matrix gain sequence $\{G_{n+1}\}$ that will result in minimum asymptotic covariance Σ_θ. It is straightforward to *Zap* our relative Q-learning algorithm, resulting in a further variance reduction.

We close with three open problems:

(i) Choice of δ. The choice $\delta = 1$ serves our purpose of uniformly bounding the asymptotic variance. Perhaps a larger δ will result in better transient behavior?

(ii) Optimizing μ (in terms of variance and transients).

(iii) Extensions outside of the tabular setting.

Acknowledgements

We thank the anonymous reviewers and the Associate Editor for valuable feedback. Financial support from ARO award W911NF2010055, and National Science Foundation award EPCN 1935389 is gratefully acknowledged.

References

[1] J. N. Tsitsiklis and B. Van Roy, "An analysis of temporal-difference learning with function approximation," *IEEE Trans. Automat. Control*, vol. 42, no. 5, pp. 674–690, 1997.

[2] V. S. Borkar and S. P. Meyn, "The ODE method for convergence of stochastic approximation and reinforcement learning," *SIAM J. Control Optim.*, vol. 38, no. 2, pp. 447–469, 2000, (see also *IEEE CDC*, 1998).

[3] A. M. Devraj and S. P. Meyn, "Zap Q-learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

[4] ——, "Fastest convergence for Q-learning," *ArXiv e-prints*, Jul. 2017.

[5] M. J. Wainwright, "Stochastic approximation with cone-contractive operators: Sharp ℓ_∞-bounds for Q-learning," *CoRR*, vol. abs/1905.06265, 2019. [Online]. Available: http://arxiv.org/abs/1905.06265

[6] G. Qu and A. Wierman, "Finite-time analysis of asynchronous stochastic approximation and Q-learning," *arXiv preprint arXiv:2002.00260*, 2020.

[7] C. Szepesvári, "The asymptotic convergence-rate of Q-learning," in *Proceedings of the 10th International Conference on Neural Information Processing Systems*, ser. NIPS'97. Cambridge, MA, USA: MIT Press, 1997, pp. 1064–1070.

[8] M. G. Azar, R. Munos, M. Ghavamzadeh, and H. Kappen, "Speedy Q-learning," in *Advances in Neural Information Processing Systems*, 2011.

[9] M. J. Wainwright, "Variance-reduced q-learning is minimax optimal," *arXiv preprint arXiv:1906.04697*, 2019.

[10] A. L. Strehl, L. Li, E. Wiewiora, J. Langford, and M. L. Littman, "PAC model-free reinforcement learning," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 881–888.

[11] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[12] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 1995, vol. 1.

[13] ——, *Dynamic Programming and Optimal Control*, 4th ed. Athena Scientific, 2012, vol. 2.

[14] A. Benveniste, M. Métivier, and P. Priouret, *Adaptive algorithms and stochastic approximations*. Springer, 2012.

[15] H. J. Kushner and G. G. Yin, *Stochastic approximation algorithms and applications*, ser. Applications of Mathematics (New York). New York: Springer-Verlag, 1997, vol. 35.

[16] V. Konda, "Actor-critic algorithms," Ph.D. dissertation, Massachusetts Institute of Technology, 2002.

[17] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint (2nd ed., to appear)*. Delhi, India and Cambridge, UK: Hindustan Book Agency, 2020.

[18] S. P. Meyn and R. L. Tweedie, *Markov chains and stochastic stability*, 2nd ed. Cambridge: Cambridge University Press, 2009, published in the Cambridge Mathematical Library. 1993 edition online.

[19] H. Robbins and S. Monro, "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951.

[20] D. Ruppert, "A Newton-Raphson version of the multivariate Robbins-Monro procedure," *The Annals of Statistics*, vol. 13, no. 1, pp. 236–245, 1985. [Online]. Available: http://www.jstor.org/stable/2241156

[21] S. Chen, A. M. Devraj, A. Bušić, and S. Meyn, "Explicit mean-square error bounds for Monte-Carlo and linear stochastic approximation," in *Proceedings of AISTATS*, S. Chiappa and R. Calandra, Eds., vol. 108. PMLR, 2020, pp. 4173–4183. [Online]. Available: http://proceedings.mlr.press/v108/chen20e.html

[22] S. Chen, A. Devraj, V. Borkar, I. Kontoyiannis, and S. Meyn, "The ODE method for asymptotic statistics in stochastic approximation and reinforcement learning," *Submitted for publication*, 2021.

[23] A. Dembo and O. Zeitouni, *Large Deviations Techniques And Applications*, 2nd ed. New York: Springer-Verlag, 1998.

[24] I. Kontoyiannis and S. P. Meyn, "Spectral theory and limit theorems for geometrically ergodic Markov processes," *Ann. Appl. Probab.*, vol. 13, pp. 304–362, 2003.

[25] A. Mokkadem and M. Pelletier, "The compact law of the iterated logarithm for multivariate stochastic approximation algorithms," *Stochastic analysis and applications*, vol. 23, no. 1, pp. 181–203, 2005.

[26] V. Koval and R. Schwabe, "A law of the iterated logarithm for stochastic approximation procedures in d-dimensional Euclidean space," *Stochastic processes and their applications*, vol. 105, no. 2, pp. 299–313, 2003.

[27] L. Devroye, L. Györfi, and G. Lugosi, *A probabilistic theory of pattern recognition*. Springer Science & Business Media, 2013, vol. 31.

[28] S. M. Kakade, "On the sample complexity of reinforcement learning," Ph.D. dissertation, University of London, 2003.

[29] T. Lattimore, M. Hutter, P. Sunehag *et al.*, "The sample-complexity of general reinforcement learning," in *Proceedings of the 30th International Conference on Machine Learning*. Journal of Machine Learning Research, 2013.

[30] M. G. Azar, R. Munos, and B. Kappen, "On the sample complexity of reinforcement learning with a generative model," *arXiv preprint arXiv:1206.6461*, 2012.

[31] E. Even-Dar and Y. Mansour, "Learning rates for Q-learning," *Journal of Machine Learning Research*, vol. 5, no. Dec, pp. 1–25, 2003.

[32] P. W. Glynn and D. Ormoneit, "Hoeffding's inequality for uniformly ergodic Markov chains," *Statistics and Probability Letters*, vol. 56, pp. 143–146, 2002.

[33] I. Kontoyiannis, L. A. Lastras-Montaño, and S. P. Meyn, "Relative entropy and exponential deviation bounds for general Markov chains," in *Proc. of the IEEE International Symposium on Information Theory*, Sept. 2005, pp. 1563–1567.

[34] R. Srikant and L. Ying, "Finite-time error bounds for linear stochastic approximation and TD learning," *CoRR*, vol. abs/1902.00923, 2019. [Online]. Available: http://arxiv.org/abs/1902.00923

[35] Z. Chen, S. T. Maguluri, S. Shakkottai, and K. Shanmugam, "Finite-sample analysis of stochastic approximation using smooth convex envelopes," *arXiv preprint arXiv:2002.00874*, 2020.

[36] E. Bolthausen, "The Berry-Esseen theorem for functionals of discrete Markov chains," *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, vol. 54, no. 1, pp. 59–73, 1980. [Online]. Available: https://doi.org/10.1007/BF00535354

[37] B. Kloeckner, "Effective Berry–Esseen and concentration bounds for markov chains with a spectral gap," *Ann. Appl. Probab.*, vol. 29, no. 3, pp. 1778–1807, 06 2019. [Online]. Available: https://doi.org/10.1214/18-AAP1438

[38] A. M. Devraj, A. Bušić, and S. Meyn, "Zap Q-Learning – a user's guide," in *Proc. of the Fifth Indian Control Conference*, January 9-11 2019.

[39] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA: MIT Press. On-line edition at http://www.cs.ualberta.ca/~sutton/book/the-book.html, 2018.

[40] S. P. Meyn and A. Surana, "TD-learning with exploration," in *50th IEEE Conference on Decision and Control, and European Control Conference*, Dec 2011, pp. 148–155.

[41] G. A. Rummery and M. Niranjan, "On-line Q-learning using connectionist systems," Cambridge Univ., Dept. Eng., Cambridge, U.K. CUED/F-INENG/, Technical report 166, 1994.

[42] C. Szepesvári, *Algorithms for Reinforcement Learning*, ser. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2010.

[43] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Mach. Learn.*, vol. 3, no. 1, pp. 9–44, 1988.

[44] J. N. Tsitsiklis and B. Van Roy, "Optimal stopping of Markov processes: Hilbert space theory, approximation algorithms, and an application to pricing high-dimensional financial derivatives," *IEEE Trans. Automat. Control*, vol. 44, no. 10, pp. 1840–1851, 1999. [Online]. Available: http://dx.doi.org/10.1109/9.793723

[45] D. Choi and B. Van Roy, "A generalized Kalman filter for fixed point approximation and efficient temporal-difference learning," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 16, no. 2, pp. 207–239, 2006.

[46] H. Yu and D. P. Bertsekas, "Q-learning and policy iteration algorithms for stochastic shortest path problems," *Annals of Operations Research*, vol. 208, no. 1, pp. 95–132, 2013. [Online]. Available: http://dx.doi.org/10.1007/s10479-012-1128-z

[47] P. G. Mehta and S. P. Meyn, "Q-learning and Pontryagin's minimum principle," in *Proc. of the IEEE Conf. on Dec. and Control*, Dec. 2009, pp. 3598–3605.

[48] C. J. C. H. Watkins and P. Dayan, "*Q*-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[49] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, King's College, Cambridge, Cambridge, UK, 1989.

[50] V. S. Borkar and K. Soumyanath, "An analog scheme for fixed point computation. I. Theory," *IEEE Trans. Circuits Systems I Fund. Theory Appl.*, vol. 44, no. 4, pp. 351–355, 1997.

[51] A. M. Devraj, A. Bušić, and S. Meyn, "Fundamental design principles for reinforcement learning algorithms," in *Handbook on Reinforcement Learning and Control*, K. G. Vamvoudakis, Y. Wan, F. L. Lewis, and D. Cansever, Eds. Springer, 2021.

[52] J. Tsitsiklis, "Asynchronous stochastic approximation and *Q*-learning," *Machine Learning*, vol. 16, pp. 185–202, 1994.

[53] J. Abounadi, D. Bertsekas, and V. S. Borkar, "Learning algorithms for Markov decision processes with average cost," *SIAM Journal on Control and Optimization*, vol. 40, no. 3, pp. 681–698, 2001.

[54] L. C. Baird, "Reinforcement learning in continuous time: Advantage updating," in *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, vol. 4. IEEE, 1994, pp. 2448–2453.

APPENDIX

A. Convergence Rate of Nonlinear Stochastic Approximation

Proof of Prop. II.2. Part (i) of the Proposition follows from the main result of [17, Ch. 7].

The proof of (ii) uses similar ideas as in [21]. For simplicity we normalize so that $\theta^* = 0$, and take $g = 1$ so that $\alpha_n = 1/n$.

The proof proceeds by contradiction: Suppose that $n^{2\varrho_1}\mathsf{E}[\|\tilde{\theta}_n\|^2]$ is bounded in n for some $\varrho_1 > \varrho_0$, and consequently $n^{2\varrho}\mathsf{E}[\|\tilde{\theta}_n\|^2]$ tends to zero as $n \to \infty$, for any $\varrho_1 > \varrho > \varrho_0$. We then use the new definition $W_n = n^{\varrho}\theta_n$, and denote, for a fixed $\theta \in \mathbb{R}^d$,

$$\overline{f}_n(\theta) = (n+1)^{\varrho}\overline{f}(n^{-\varrho}\theta)\,, \quad \Upsilon_n = \alpha_n n^{\varrho}\Delta_n$$

On multiplying each side of (1) by $(n+1)^{\varrho}$ we obtain

$$W_{n+1} = W_n + \alpha_{n+1}[\varrho_n W_n + \overline{f}_n(W_n)] + \Upsilon_{n+1}$$

where $\varrho_n = \varrho + o(1)$ appears through the Taylor series approximation $(n+1)^{\varrho} = n^{\varrho} + \varrho\alpha_{n+1}n^{\varrho} + o(\alpha_{n+1})$.

Under the *assumption* that $n^{2\varrho_1}\mathsf{E}[\|\tilde{\theta}_n\|^2]$ is bounded in n, it follows that

$$\lim_{n\to\infty} \mathsf{E}[\|\overline{f}_n(W_n) - \overline{f}_0(W_n)\|^2] = 0$$

and from this we obtain the approximately linear recursion for $\Sigma_n^W = \mathsf{E}[W_n W_n^{\intercal}]$:

$$\Sigma_{n+1}^W = \Sigma_n^W + \alpha_{n+1}[(\varrho + A)\Sigma_n^W + \Sigma_n^W(\varrho + A)^{\intercal} + \mathcal{E}_n] + \Sigma_{n+1}^{\Upsilon}$$

where the vanishing sequence $\{\mathcal{E}_n\}$ is composed of three approximations: replacing ϱ_n by ϱ, the replacement of $\overline{f}_n$ by $\overline{f}_0$, and the final term:

$$\alpha_{n+1}^2\mathsf{E}[(\varrho_n W_n + \overline{f}_n(W_n))(\varrho_n W_n + \overline{f}_n(W_n))^{\intercal}]$$

Denote $\sigma_n^2 = n^{2\varrho}\mathsf{E}[|\nu^{\intercal}W_n|^2] = \nu^{\intercal}\Sigma_n^W\nu$, a vanishing sequence that evolves according to the recursion

$$\sigma_{n+1}^2 = \sigma_n^2 + \alpha_{n+1}[2(\varrho - \varrho_0)\sigma_n^2 + \nu^{\intercal}\mathcal{E}_n\nu] + \nu^{\intercal}\Sigma_{n+1}^{\Upsilon}\nu$$

As in [21], this can be regarded as a deterministic SA recursion that is unstable under our assumption that $\varrho - \varrho_0 > 0$. An ODE approximation, along with the fact that $\nu^{\intercal}\Sigma_{n+1}^{\Upsilon}\nu > 0$ for at least one n, implies that $\sigma_n^2 \to \infty$ as $n \to \infty$ under this assumption. This contradiction completes the proof. ☐

B. ODE Approximation of Q-learning

Proof of Lemma IV.1. We prove (i) only, since the (ii) is identical. Recall the definition of the ODE (53): $\frac{d}{dt}q_t = \overline{f}(q_t)$, where for any $q\colon \mathsf{X}\times\mathsf{U} \to \mathbb{R}$, and $1 \le i \le d$,

$$\overline{f}_i(q) = \mathsf{E}\big[\{c(X_n,U_n) + \gamma\underline{q}(X_{n+1}) - q(X_n,U_n)\}\psi_i(X_n,U_n)\big]$$

Substituting $\underline{q}(X_{n+1}) = q(X_{n+1},\phi^*(X_{n+1}))$ for $\|q - Q^*\| < \varepsilon$ implies (i), (recalling the tabular basis):

$$\begin{aligned}\overline{f}_i(q) &= \mathsf{E}\big[c(X_n,U_n)\psi_i(X_n,U_n)\big] \\ &\quad + \mathsf{E}\big[\psi_i(X_n,U_n)\{\gamma q(X_{n+1},\phi^*(X_{n+1})) - q(X_n,U_n)\}\big] \\ &= \varpi(x^i,u^i)c(x^i,u^i) \\ &\quad + \varpi(x^i,u^i)\{\gamma\sum_j P_{u^i}(x^i,x^j)q(x^j,\phi^*(x^j)) - q(x^i,u^i)\}\end{aligned}$$

C. Convergence Analysis of Relative Q-learning ☐

Prop. A.1 is established here, which follows from Props. A.3 and A.4 below.

Proposition A.1. *The ODE (73) is globally asymptotically stable, with unique equilibrium H^*.*

For any $H : \mathsf{X}\times\mathsf{U} \to \mathbb{R}$, the span semi-norm is denoted

$$\|H\|_S := \max_{x,u} H(x,u) - \min_{x,u} H(x,u) \tag{102}$$

Th operator $\widetilde{T}$ defined in (75) is a γ-contraction [11]:

Lemma A.2. *For any $0 \le \gamma < 1$, the operator $\widetilde{T}$ is a γ-contraction: For any $H : \mathsf{X}\times\mathsf{U} \to \mathbb{R}$ and $H' : \mathsf{X}\times\mathsf{U} \to \mathbb{R}$,*

$$\|\widetilde{T}H - \widetilde{T}H'\|_S \le \gamma\|H - H'\|_S \tag{103}$$

The contraction property is used next to prove the stability of the ODE (73) in the span semi-norm.

For each $t \geq 0$, define $\tilde{h}_t := h_t - H^*$, where H^* is the unique solution to the fixed point equation (65). Define

$$\phi_t^* := \phi^{(\kappa)},\ \kappa = \min\{i : \phi^{(i)}(x) \in \arg\min_u h_t(x,u)\}$$

Prop. A.3 establishes exponential convergence of h_t to H^* in the span semi-norm, which implies the same rate of convergence for $H^*(x,\phi_t^*(x))$ to $H^*(x,\phi^*(x))$.

Proposition A.3. *For any $0 \leq \gamma < 1$, and some $K < \infty$,*

$$\|\tilde{h}_t\|_S \leq e^{-(1-\gamma)t}\|\tilde{h}_0\|_S \quad (104a)$$

$$\|H^*(x,\phi_t^*(x)) - H^*(x,\phi_t^*(x))\| \leq Ke^{-(1-\gamma)t}\|\tilde{h}_0\| \quad (104b)$$

Proof. By the variations of constants formula, and using the notation (75), the solution h_t to (73) satisfies:

$$h_t(x\,,u) = h_0(x\,,u)e^{-t} + \int_0^t e^{-(t-s)}(\widetilde{T}h_s)(x\,,u)\,ds$$

Subtracting $H^*(x\,,u)$ from both sides, and using the fact that $H^*(x\,,u) = (\widetilde{T}H^*)(x\,,u)$, we obtain

$$\tilde{h}_t(x\,,u) = \tilde{h}_0(x,u)e^{-t} + \int_0^t e^{-(t-s)}\Big[(\widetilde{T}h_s)(x\,,u) - (\widetilde{T}H^*)(x\,,u)\Big]ds$$

The following inequalities are then immediate

$$\max_{x\,,u}\tilde{h}_t(x\,,u) \leq \max_{x\,,u}\tilde{h}_0(x\,,u)e^{-t} + \int_0^t e^{-(t-s)}\max_{x\,,u}\Big[(\widetilde{T}h_s)(x\,,u) - (\widetilde{T}H^*)(x\,,u)\Big]ds \quad (105a)$$

$$\min_{x\,,u}\tilde{h}_t(x\,,u) \geq \min_{x\,,u}\tilde{h}_0(x\,,u)e^{-t} + \int_0^t e^{-(t-s)}\min_{x\,,u}\Big[(\widetilde{T}h_s)(x\,,u) - (\widetilde{T}H^*)(x\,,u)\Big]ds \quad (105b)$$

Subtracting (105b) from (105a),

$$\begin{aligned}\|\tilde{h}_t\|_S &\leq e^{-t}\|\tilde{h}_0\|_S + \int_0^t e^{-(t-s)}\|\widetilde{T}h_s - \widetilde{T}H^*\|_S\,ds \\ &\leq e^{-t}\|\tilde{h}_0\|_S + \gamma\int_0^t e^{-(t-s)}\|\tilde{h}_s\|_S\,ds\end{aligned} \quad (106)$$

where the second inequality follows from (103). Therefore,

$$e^t\|\tilde{h}_t\|_S \leq \|\tilde{h}_0\|_S + \gamma\int_0^t e^s\|\tilde{h}_s\|_S\,ds$$

Applying the Grönwall's inequality completes the proof of (104a); (104b) follows from (69) and (104a). □

Define for each $t \geq 0$

$$r_t := \langle \mu\,, \tilde{h}_t\rangle \quad (107)$$

Prop. A.3 (in particular, Eq. (104a)) implies $h_t \to H^*$ exponentially fast, in the span-semi-norm: for some $K < \infty$,

$$\tilde{h}_t = \mathbb{1}\cdot r_t + \varepsilon_t^s \quad \|\varepsilon_t^s\| \leq Ke^{-(1-\gamma)t}\|\tilde{h}_0\|_S \quad (108)$$

To establish global exponential stability of the ODE (73), it is sufficient to show that $r_t \to 0$ exponentially fast.

Proposition A.4. *For any $\gamma < 1$, and $\delta > 0$, the function r_t defined in (107) satisfies, for some $K < \infty$,*

$$|r_t| \leq e^{-(1-\gamma+\delta)}|r_0| + Ke^{-(1-\gamma)t}\|\tilde{h}_0\|$$

Proof. Differentiating both sides of (107), and using (73),

$$\begin{aligned}\tfrac{d}{dt}r_t = \langle\mu\,,\tfrac{d}{dt}h_t\rangle &= \langle\mu\,,\widetilde{T}h_t - h_t\rangle \\ &= \langle\mu\,,\widetilde{T}h_t - h_t - \widetilde{T}H^* + H^*\rangle\end{aligned} \quad (109)$$

where we have used the fact that $H^* = \widetilde{T}H^*$.

Using (108) and (104b), the non-linear term on the right hand side of (109) admits the approximation,

$$\begin{aligned}&\big\langle\mu\,,\widetilde{T}h_t - \widetilde{T}H^*\big\rangle \\ &= \gamma\sum_{x,u,x'}\mu(x,u)P_u(x,x')\Big[h_t\big(x',\phi_t^\varepsilon(x')\big) - H^*\big(x',\phi^*(x')\big)\Big] \\ &\qquad - \delta\cdot\langle\mu\,,\tilde{h}_t\rangle \\ &= \gamma\sum_{x,u,x'}\mu(x,u)P_u(x,x')\Big[H^*\big(x',\phi_t^\varepsilon(x')\big) - H^*\big(x',\phi^*(x')\big) \\ &\qquad + \varepsilon_t^s\big(x',\phi_t^\varepsilon(x')\big)\Big] + \big(\gamma-\delta\big)\cdot r_t \\ &= \big(\gamma-\delta\big)\cdot r_t + \varepsilon_t^r\end{aligned}$$

where $|\varepsilon_t^r| \leq Ke^{-(1-\gamma)t}\|\tilde{h}_0\|$ for some $K < \infty$. Substituting this into (109) completes the proof:

$$\begin{aligned}\tfrac{d}{dt}r_t &= \big(\gamma-\delta-1\big)\cdot r_t + \varepsilon_t^r, \\ r_t &= e^{-(1-\gamma+\delta)}\cdot r_0 + \int_0^t e^{-(1-\gamma+\delta)\tau}\cdot\varepsilon_{t-\tau}^r\,d\tau\end{aligned}$$

□

Adithya M. Devraj is a postdoctoral research fellow at Stanford University, where he is working with Prof. Benjamin Van Roy. He received his M.S. (2016) and Ph.D. (2019) degrees in Electrical and Computer Engineering from the University of Florida, Gainesville, FL, USA, where he worked with Prof. Sean P. Meyn on theory of stochastic approximation and reinforcement learning algorithms. Before arriving at Florida in 2014, he received his B.E. from PES Institute of Technology, Bangalore, India, in 2013, and worked at the Signal Processing for Communication lab, Indian Institute of Science, Bangalore, India, between 2013 and 2014 with Prof. Chandra R. Murthy. He has also held visiting positions at Inria, Paris, and the Simon's Institute for the Theory of Computing, UC Berkeley.

Sean P. Meyn (S'85-M'87-SM'95-F'02) received the Ph.D. degree in electrical engineering from McGill University, Montreal, QC, Canada, in 1987 (with Prof. P. Caines). He held a two year postdoctoral fellowship at the Australian National University, Canberra, Act, Australia, and was a Professor at the University of Illinois from 1989 to 2011. Since January 2012, he has been a Professor and has held the Robert C. Pittman Eminent Scholar Chair in electrical and computer engineering at the University of Florida, Gainesville, FL, USA. He is the coauthor with Richard Tweedie of the monograph Markov Chains and Stochastic Stability, (Springer-Verlag, 1993). Prof. Meyn jointly received the 1994 ORSA/TIMS Best Publication in Applied Probability Award with Tweedie. He and Prof. Bušić share a 2015 Google Research Award: to foster collaboration on demand dispatch for renewable energy integration.