Zap Q-Learning: Technical Survey

Naman Aggarwal (160010058)
Systems and Control Engineering, IIT Bombay
naman_agg@iitb.ac.in

June 8, 2020

Abstract

This technical report is a survey on a series of work done by Adithya Devraj and Sean Meyn on 'Zap Q-Learning' [5] [6], an improvement over Watkin's original Q-Learning algorithm. It is a matrix-gain algorithm designed so that it's asymptotic variance is optimal. This is made possible by a two time-scale update equation for the matrix-gain sequence, inspired from a Zap version of the standard Newton-Raphson algorithm in stochastic settings. The goal of this study is to shed clarity on recursive stochastic estimation techniques and related important considerations such as rate of convergence, covariance of the estimates etc.
Keywords: Reinforcement learning, Q-learning, Stochastic optimal control

1 Introduction

The slow convergence of algorithms such as TD-learning and Q-learning has been characterized in literature. The poor performance of Q-learning was first quantified in [14] and since then, many papers have appeared with proposed improvements [7] [8]. A general theory has been presented in [12] for stochastic approximation algorithms.

In recursive stochastic approximation techniques, the update equation for the parameter estimates can be expressed as,

$$\theta_{n+1} = \theta_n + \alpha_n[\overline{f}(\theta_n) + \Delta_{n+1}] \tag{1}$$

in which $\{\alpha_n\}$ is a positive gain sequence, and $\{\Delta_n\}$ is a martingale difference sequence. In the above formulation, we can say that a martingale noise sequence drives the estimation procedure. We can also have a Markovian model in which the estimation procedure is driven by the transition mechanism of a 'noise' or 'state' Markov process (see [10]) as illustrated below,

$$\theta_{n+1} = \theta_n + \alpha_n f(\theta_n, X_{n+1}) \tag{2}$$

where $\{X_n\}$ is a Markov state process. For Markovian models, the usual transformation used to obtain a representation similar to (1) results in an error sequence $\{\Delta_n\}$, that is the sum of a martingale difference sequence and a telescoping sequence [10]. It is the telescoping sequence which lends additional complexity and prevents the easy analysis of Markovian models. This gap in research literature carries over to the general theory of Markov chains. While for i.i.d. and martingale difference sequences, we have the concentration bounds of Hoeffding and Bennett respectively, extensions of such bounds to Markovian models is an active area of research [13].

We are especially concerned with the asymptotic behaviour of our iteration scheme (1) and want the limiting distribution of the sequence $\{\theta_n\}$ to be centered around θ^* and have low covariance. The Central Limit Theorem which holds under very general conditions comes in handy to study the asymptotic behaviour [1, 2, 9]. For a typical stochastic estimation procedure, denoting $\tilde{\theta}_n = \theta_n - \theta^* : n \geq 0$ to be the error sequence, under general conditions the scaled sequence $\{\sqrt{n}\theta_n : n \geq 1\}$ converges in distribution to a Gaussian distribution, $N(0, \Sigma_\theta)$. Typically, the scaled covariance is also convergent

$$\Sigma_\theta = \lim_{n\to\infty} nE[\tilde{\theta}_n\tilde{\theta}_n^T] \tag{3}$$

The asymptotic covariance Σ_θ has a simple representation as the solution to a Lyapunov equation as will be discussed in the upcoming sections. It also gives a good approximation for finite time performance of the iteration scheme, since the CLT is accurate for reasonable values of n.

We start our discussion with the Stochastic Newton-Raphson algorithm in the linear setting for clarity on how to design a gain-matrix for the stochastic approximation recursion resulting in an optimal covariance of the estimates. The discussion then continues by introducing Zap SNR which is a two time-step algorithm in which the gain-matrix is also recursively estimated on-the-go along with the parameter sequence. Discussion on Watkin's Q-learning algorithm ensues and how the Zap-SNR algorithm is adapted as a solution methodolgy for the fixed point equation corresponding to Watkin's learning giving rise to the Zap Q-learning algorithm with improved asymptotic covariance guarantees.

2 Stochastic Newton Raphson

2.1 Fundamentals

The goal of stochastic approximation is to compute the solution $\overline{f}(\theta^*) = 0$ for a function $\overline{f} : R^d \to R^d$. If the function is easily evaluated and satisfies some conditions, we can use the Newton-Raphson algorithm as follows

$$\theta_{n+1} = \theta_n + G_n \overline{f}(\theta_n), \quad G_n^{-1} = -\nabla \overline{f}(\theta_n) \tag{4}$$

Stochastic approximation can be employed when we do not have access to true the function value at parameter θ_n but instead a noisy realization of the same. We can therefore right $\overline{f}$ as $\overline{f}(\theta) = E[f(\theta, \xi)]$, where $f : R^d \times R^m \to R^d$ and ξ is a random variable with some distribution. The standard stochastic approximation algorithm can then be employed,

$$\theta_{n+1} = \theta_n + \alpha_n f(\theta_n, \xi_{n+1}) \tag{5}$$

If $\{\xi_n\}$ is a Markov state process, we can assume for simplicity that ξ which defines $\overline{f}(\theta) = E[f(\theta, \xi)]$, is the stationary realization of the Markov process. This model occurs very frequently in reinforcement learning. It is always assumed that the scalar gain sequence $\{\alpha_n\}$ is non-negative and defines,

$$\sum \alpha_n = \infty, \quad \sum \alpha_n^2 < \infty$$

While convergent under general conditions, we can improve the rate of convergence of (5) by the introduction of a matrix-gain sequence. This is illustrated first in the simple linear setting.

2.2 Optimal covariance for linear stochastic approximation

Consider the situation where you want the solution θ^* to the equation $A\theta - b = 0$, where $A = E[A(\xi)]$ is a $d \times d$ matrix and $b = E[b(\xi)]$ is a $d \times 1$ vector, ξ is the steady state distribution of some Markov chain. What we have available are noisy samples: $A_n = A(\xi_n)$ and $b_n = b(\xi_n)$ for the Markov state process $\{\xi_n\}$. A and b therefore represent the steady state means of the processes $\{A_n\}$ and $\{b_n\}$ respectively. We employ the following recursive estimation scheme,

$$\theta_{n+1} = \theta_n + \alpha_{n+1}[A_{n+1}\theta_n - b_{n+1}] \tag{6}$$

It is assumed throughout this section that A is Hurwitz, that is the real part of each eigenvalue is negative. Under this assumption and subject to mild conditions on ξ, it is known that $\{\theta_n\}$ converges with probability one to $\theta^* = A^{-1}b$ [1, 2, 9]. The convergence of recursion (6) will be assumed henceforth. We assume that the gain sequence is given by $\alpha_n = 1/n,\ n \geq 1$. Defining the error sequence as $\{\tilde{\theta}_n = \theta_n - \theta^*\}$, therefore error covariance at timestep n is $\Sigma_n = nE[\tilde{\theta}_n \tilde{\theta}_n^T]$. We can obtain a relation between Σ_{n+1} and Σ_n by substituting (6) into $\Sigma_{n+1} = (n+1)E[\tilde{\theta}_{n+1}\tilde{\theta}_{n+1}^T]$. Using this relation and taking limit $n \to \infty$, following relation for the asymptotic covariance is obtained,

$$(A + \frac{1}{2}I)\Sigma_\theta + \Sigma_\theta(A + \frac{1}{2}I)^T + \Sigma_\Delta = 0 \tag{7}$$

A solution is guaranteed only if each eigenvalue of A has real part strictly less than -1/2. This is a stronger condition than the Hurwitz condition and if violated, the asymptotic covariance under general conditions is infinity, meaning our estimation procedure has revealed nothing useful about θ^*.

Matrix Σ_Δ used in (7) is obtained as follows: based on (6), the error sequence $\{\tilde{\theta}_n = \theta_n - \theta^*\}$ evolves according to a deterministic linear system driven by "noise":

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

in which Δ_{n+1} is,

$$\Delta_{n+1} = \tilde{A}_{n+1}\theta^* - \tilde{b}_{n+1} + \tilde{A}_{n+1}\theta_{n+1} \tag{8}$$

with $\tilde{A}_{n+1} = A_{n+1} - A,\ \tilde{b}_{n+1} = b_{n+1} - b$. It is assumed that the CLT holds for sample averages of the noise sequence:

$$\frac{1}{\sqrt{N}}\sum_{n=1}^{N}\Delta_n \to N(0, \Sigma_\Delta),\ N \to \infty \tag{9}$$

where the limit is in distribution. This is a mild requirement when ξ is Markovian [11]. A finite asymptotic covariance can be guaranteed by increasing the gain: choose $\alpha_n = g/n$ in (6) with $g > 0$ sufficiently large so that the eigenvalues of gA satisfy the required bound. We can introduce a matrix gain as follows:

$$\theta_{n+1} = \theta_n + \frac{1}{n+1}G(A_{n+1}\theta_{n+1} - b_{n+1}) \tag{10}$$

in which G is a $d \times d$ matrix. Provided the matrix GA satisfies the eigenvalue bound, the corresponding asymptotic covariance Σ_θ^G is finite and solves a modified Lyapunov equation:

$$(GA + \frac{1}{2}I)\Sigma_\theta^G + \Sigma_\theta^G(GA + \frac{1}{2}I)^T + G\Sigma_\Delta G^T = 0 \tag{11}$$

The choice $G^* = -A^{-1}$ is analogous to the gain used in Newton-Raphson algorithm. With this choice, the asymptotic covariance is finite and given by,

$$\Sigma^* = A^{-1}\Sigma_\Delta {A^{-1}}^T$$

This is the optimal choice for the gain matrix. For any other gain matrix G,

$$\Sigma_\theta^G \geq \Sigma^*$$

That is, the difference $\Sigma_\theta^G - \Sigma^*$ is positive semi-definite. Our findings can be summarized in the theorem below.

Theorem 2.1 *Suppose that eigenvalues of GA lie in the strict left half plane, and that the noise sequence satisfies CLT (9) with finite covariance Σ_Δ. Then, the stochastic approximation recursion defined in (10) is convergent, and the following also hold:*

1. If all the eigenvalues of GA satisfy Re$(\lambda) < -1/2$, then the corresponding asymptotic covariance Σ_θ^G is finite, and can be obtained as the solution to the Lyapunov equation (11).

2. For any matrix gain G, the asymptotic covariance admits the lower bound

$$\Sigma_\theta^G \geq \Sigma^* := A^{-1}\Sigma_\Delta {A^{-1}}^T$$

This lower bound is achieved using $G^* := -A^{-1}$.

Thm 2.1 inspires improved algorithms in many settings, the first one being Stochastic Newton-Raphson.

2.3 Stochastic Newton-Raphson

This algorithm is obtained by estimating the mean A simultaneously with the estimation of θ^*.

$$\begin{aligned}\theta_{n+1} &= \theta_n - \alpha_n \hat{A}_{n+1}^{-1}[A_{n+1}\theta_n - b_{n+1}] \\ \hat{A}_{n+1} &= \hat{A}_n + \alpha_n[A_{n+1} - \hat{A}_{n+1}], \quad \alpha_{n+1} = \frac{1}{n+1}\end{aligned} \tag{12}$$

where θ_0 and $\hat{A}_1$ are initial conditions. If the steady state mean A is invertible, then $\hat{A}_n$ is invertible for all n sufficiently large. Supposing that $\hat{A}_n$ is invertible for each $n \geq 1$ and for the step-size sequence $\{\alpha_n\}$ as defined $(= \frac{1}{n+1})$, the recursion (12) results in a simple representation of the parameter estimates: $\theta_n = \hat{A}_n^{-1}\hat{b}_n$, where $\hat{A}_n = \frac{1}{n}\sum_{i=1}^n A_i,\ \hat{b}_n = \frac{1}{n}\sum_{i=1}^n b_i,\ n \geq 1$. Therefore, the SNR algorithm is consistent whenever the Law of Large Numbers holds for the sequence $\{A_n, b_n\}$. Under the assumptions of Thm 2.1, the resulting asymptotic covariance is identical to what would be obtained with the constant matrix gain $G^* = -A^{-1}$.

Algorithm design in this linear setting is simplified in part because $\overline{f}$ is an affine function of θ, so that the gain G_n appearing in the standard Newton-Raphson (4) algorithm does not depend upon the parameter estimates $\{\theta_k\}$. An ODE analysis of the SNR algorithm suggests the following dynamics:

$$\begin{aligned}\frac{d}{dt}\theta_t &= -A_t^{-1}[A\theta_t - b] \\ \frac{d}{dt}A_t &= -A_t + A\end{aligned} \tag{13}$$

While evidently A_t converges to A exponentially fast in the linear model, with a poor initial condition we might expect poor transient behavior. In extending the SNR algorithm to a nonlinear stochastic approximation algorithm, an ODE approximation of the form (13) will be possible under general conditions, but the matrix A will depend on θ. And, just as in the linear model, the continuous time system looks very different from the deterministic Newton-Raphson recursion (4). The next class of algorithms are designed so that the associated ODE more closely matches the deterministic recursion.

3 Zap Stochastic Newton-Raphson

3.1 Introduction

This is a two time-step algorithm with a higher step-size for the Matrix recursion. For the linear setting, it is defined by a variant of (12):

$$\begin{aligned}\theta_{n+1} &= \theta_n - \alpha_{n+1}\hat{A}_{n+1}^{-1}[A_{n+1}\theta_n - b_{n+1}] \\ \hat{A}_{n+1} &= \hat{A}_n + \gamma_{n+1}[A_{n+1} - \hat{A}_n]\end{aligned} \tag{14}$$

It is different from the original SNR algorithm because of the two time-scale construction. The second step-size sequence $\{\gamma_{n+1}\}$ is non-negative, satisfies the usual Robbins-Monro condition and also,

$$\lim_{n\to\infty} \frac{\alpha_n}{\gamma_n} = 0 \tag{15}$$

Again, $\alpha_n = \frac{1}{n},\ n \geq 1$. The asymptotic covariance is again optimal. The ODE associated with the sequence $\{\theta_n\}$ is far simpler, and exactly matches the usual Newton-Raphson dynamics:

$$\frac{d}{dt}\theta_t = -\theta_t + A^{-1}b \tag{16}$$

A key point to note here is that the Zap version of the SNR algorithm plays a significant role in analysis as well as in performance improvement of general non-linear function approximation problems which we discuss in the next sub-section.

3.2 Zap SNR for non-linear stochastic approximation

Consider a stochastic approximation algorithm of the form (5) with $\overline{f}(\theta) = E[f(\theta, \xi)]$, a non-linear function of the parameter vector θ. The ODE of the two algorithms: SNR and Zap-SNR is significantly different and this difference extends to the rate of convergence of the stochastic recursion. The SNR algorithm for our non-linear setup is essentially the same as (12):

$$\begin{aligned} \theta_{n+1} &= \theta_n - \alpha_{n+1}\hat{A}_{n+1}^{-1} f(\theta_n, \xi_{n+1}) \\ \hat{A}_{n+1} &= \hat{A}_n + \alpha_{n+1}[\nabla f(\theta_n, \xi_{n+1}) - \hat{A}_{n+1}], \quad \alpha_{n+1} = \frac{1}{n+1} \end{aligned} \tag{17}$$

Not that the function $\nabla f(\theta_n, \xi_{n+1})$ may or may not be easily computable and this is application specific. In the case of Q-learning with linear function approximation, though the function f is itself non-linear in θ, ∇f is readily computable.
The ODE for the pair of recursions (17) once again will be similar to (13):

$$\begin{aligned} \frac{d}{dt}\theta_t &= -A_t^{-1}\overline{f}(\theta_t) \\ \frac{d}{dt}A_t &= -\nabla\overline{f}(\theta_t) + A \end{aligned} \tag{18}$$

The Zap-SNR is a generalization of (14):

$$\begin{aligned} \theta_{n+1} &= \theta_n - \alpha_{n+1}\hat{A}_{n+1}^{-1} f(\theta_n, \xi_{n+1}) \\ \hat{A}_{n+1} &= \hat{A}_n + \gamma_{n+1}[\nabla f(\theta_n, \xi_{n+1}) - \hat{A}_n] \end{aligned} \tag{19}$$

where once again the step-size sequence $\{\gamma_n\}$ satisfies the usual Robbins-Monro conditions and (15). Similar to (16), the ODE of the algorithm is identical to the deterministic Newton-Raphson dynamics:

$$\frac{d}{dt}\theta_t = -(\nabla\overline{f}(\theta_t))^{-1}\overline{f}(\theta_t) \tag{20}$$

The general convergence and stability analysis of both (18) and (20) is open. In the next section, we see that when applied to Q-learning, the algorithms do converge under certain technical conditions.

4 Zap Q-Learning

4.1 Introduction: Q-Learning

Consider an MDP model with state space X, and action space U, cost function $c : X \times U \to \mathbb{R}$, and discount factor $\beta \in (0, 1)$. It is assumed that the state and action space are finite: denote $l = |X|$, and $l_u = |U|$, and P_u the $l \times l$ transition probability matrix, conditioned on $u \in U$. The state-action process $(\mathbf{X}, \mathbf{U})$ is adapted to the filtration $\{F_n : n \geq 0\}$ and A1 is assumed throughout:
A1: The joint process $(\mathbf{X}, \mathbf{U})$ is an irreducible Markov chain, with unique invariant pmf ϖ.
The minimal value function is the unique solution to the discounted-cost optimality equation:

$$h^*(x) = \min_{u\in U} Q^*(x, u) := \min_{u\in U}\{c(x, u) + \beta \sum_{x'\in X} P_u(x, x')h^*(x')\}, \quad x \in X$$

The Q-function solves a similar fixed point equation:

$$Q^*(x, u) = c(x, u) + \beta \sum_{x'\in X} P_u(x, x')\underline{Q}^*(x'), \quad x \in X, u \in U, \tag{21}$$

where $\underline{Q}(x) = \min_{u\in U} Q(x, u)$ for any function $Q : X \times U \to \mathbb{R}$.
Given any function $\varsigma : X \times U \to \mathbb{R}$, let $Q(\varsigma)$ denote the corresponding solution to (21) with c replaced by ς. The function $q = Q(\varsigma)$ denotes the solution to the fixed point equation,

$$q(x, u) = \varsigma(x, u) + \beta \sum_{x'\in X} P_u(x, x') \min_{u'\in U} q(x', u'), \quad x \in X, u \in U.$$

The mapping Q is a bijection on the set of real-valued functions on $X \times U$ (See [6] for proofs and discussion). It is known that Watkin's Q-learning algorithm can be regarded as a stochastic approximation method [3, 15] to obtain the solution $\theta^* \in \mathbb{R}^d$ to the steady-state mean equations,

$$E[\{c(X_n, U_n) + \beta \underline{Q}^{\theta^*}(X_{n+1}) - Q^{\theta^*}(X_n, U_n)\}\zeta_n(i)] = 0, \quad 1 \le i \le d \tag{22}$$

where $\{\zeta_n\}$ are d-dimensional F_n-measurable functions and $Q^\theta = \theta^T\psi$ for basis functions $\{\psi_i : 1 \le i \le d\}$. In Watkin's algorithm $\zeta_n = \psi(X_n, U_n)$ and the basis functions are indicator functions: $\psi_k(x, u) = \mathbb{I}\{x = x^k, u = u^k\}, 1 \le i \le d$ with $d = l \times l_u$ is the total number of state-action pairs. In this special case, we identify $Q^{\theta^*} = Q^*$, and the perimeter θ is identified with the estimate Q^θ. A stochastic approximation algorithm to solve (22) coincides with Watkins' algorithm [16]:

$$\theta_{n+1} = \theta_n + \alpha_{n+1}\{c(X_n, U_n) + \beta \underline{\theta}_n(X_{n+1}) - \theta_n(X_n, U_n)\}\psi(X_n, U_n) \tag{23}$$

We study the *associated limiting ODE* to analyze the convergence of our stochastic approximation algorithm. For (23), denoting the continuous-time approximation of $\{\theta_n\}$ to be $\{q_t\}$ and under standard Robbins-Monro conditions on the gain sequence $\{\alpha_n\}$, the associated ODE is of the form:

$$\frac{d}{dt}q_t(x, u) = \varpi(x, u)\{c(x, u) + \beta \sum_{x' \in X} P_u(x, x') \min_{u' \in U} q_t(x', u') - q_t(x, u)\} \tag{24}$$

Under **A1**, $\{q_t\}$ converges to Q^*. It is argued in [6] that the asymptotic covariance of Watkin's Q-learning algorithm is typically infinite. This conclusion is contrary to the finite-time analysis of [14]. We state the following theorem from [6].

Theorem 4.1 *Watkin's Q-learning algorithm with step-size $\alpha_n = 1/n$ is consistent under assumption A1. Suppose that in addition $\max_{x,u} \varpi(x, u) \le \frac{1}{2}(1-\beta)^{-1}$, and conditional variance of $h^*(X_t)$ is positive:*

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

Then the asymptotic covariance is infinite: $\lim_{n\to\infty} nE[\|\theta_n - \theta^*\|^2] = \infty$.

Note that the assumption $\max_{x,u} \varpi(x, u) \le \frac{1}{2}(1 - \beta)^{-1}$ is automatically satisfied for $\beta \ge \frac{1}{2}$.

4.2 Matrix-gain improvements and Zap Q-learning

Matrix gain techniques have been used in literature to speed-up the rate of convergence of stochastic approximation algorithms. There also has been work on improving the rate of convergence of Q-learning in literature (see [4]). The general $G - Q(\lambda)$ algorithm based on a sequence of $d \times d$ gain matrices $\mathbf{G} = \{G_n\}$ and $\lambda \in [0, 1]$ is described as follows: For initialization $\theta_0, \zeta_0 \in \mathbb{R}^d$, the sequence of estimates are defined recursively:

$$\begin{aligned}
\theta_{n+1} &= \theta_n + \alpha_{n+1}G_{n+1}\zeta_n d_{n+1} \\
d_{n+1} &= c(X_n, U_n) + \beta \underline{Q}^{\theta_n}(X_{n+1}) - Q^{\theta_n}(X_n, U_n) \\
\zeta_{n+1} &= \lambda\beta\zeta_n + \psi(X_{n+1}, U_{n+1})
\end{aligned} \tag{25}$$

where ζ_n is the eligibility vector. A special case of such matrix-gain algorithms is Zap Q(λ)-learning: The main contributions of [5] concern a two time-scale implementation inspired by Zap SNR, for which:

$$\sum \gamma_n = \infty, \sum \gamma_n^2 < \infty \quad \text{and} \quad \lim_{n\to\infty} \frac{\alpha_n}{\gamma_n} = 0 \tag{26}$$

Through ODE analysis, it can be seen that the Zap Q-learning algorithm closely resembles an implementation of Newton-Raphson for our case. $\hat{A}_n$ more closely tracks the mean of $\{A_n\}$. Thm 4.2 summarizes the main results under **A1** and the following additional assumptions:
A2: The optimal policy ϕ^* is unique.
A3: The sequence of policies $\{\phi_n\}$ satisfy $\sum_{n=1}^\infty \gamma_n \mathbb{I}\{\phi_{n+1} \ne \phi_n\} < \infty$, a.s..

Algorithm 1 Zap Q(λ)-learning

Input: $\theta_0 \in \mathbb{R}^d, \zeta_0 = \psi(X_0, U_0), \hat{A}_0 \in \mathbb{R}^{d\times d}, n = 0, T$
repeat
  $\phi_n\left(X_{n+1}\right) := \arg\min_u Q^{\theta_n}\left(X_{n+1}, u\right)$
  $d_{n+1} := c\left(X_n, U_n\right) + \beta Q^{\theta_n}\left(X_{n+1}, \phi_n\left(X_{n+1}\right)\right) - Q^{\theta_n}\left(X_n, U_n\right)$
  $A_{n+1} := \zeta_n\left[\beta\psi\left(X_{n+1}, \phi_n\left(X_{n+1}\right)\right) - \psi\left(X_n, U_n\right)\right]^T$
  $\widehat{A}_{n+1} = \widehat{A}_n + \gamma_{n+1}\left[A_{n+1} - \widehat{A}_n\right]$
  $\theta_{n+1} = \theta_n - \alpha_{n+1}\widehat{A}_{n+1}^{-1}\zeta_n d_{n+1}$
  $\zeta_{n+1} := \lambda\beta\zeta_n + \psi\left(X_{n+1}, U_{n+1}\right)$
  $n = n + 1$
until $n \geq T$

Theorem 4.2 *Suppose that assumptions A1-A3 hold, and the gain sequences α and γ satisfy:*

$$\alpha_n = n^{-1},\ \gamma_n = n^{-\rho}, \quad n \geq 1$$

for some fixed $\rho \in (\frac{1}{2}, 1)$. Then,
(i) The parameter sequence $\{\theta_n\}$ obtained using the Zap Q-learning algorithm converges to Q^* a.s..
(ii) The asymptotic covariance (3) is minimized over all G-Q(0) matrix gain versions of the Watkin's Q-learning algorithm.

Remark: Refer to [6] for complete proof of Thm 4.2.

References

[1] A. Benveniste. M. m etivier, and p. priouret. adaptive algorithms and stochastic approximations. *Applications of Mathematics. Springer Verlag*, 1990.

[2] V. Borkar. Stochatic approximation: A dynamical systems viewpoint, hindustan book agency, new delhi, india, and cambridge uni. *Press, Cambridge, UK*, 2008.

[3] V. S. Borkar and S. P. Meyn. The ode method for convergence of stochastic approximation and reinforcement learning. *SIAM Journal on Control and Optimization*, 38(2):447–469, 2000.

[4] D. Choi and B. Van Roy. A generalized kalman filter for fixed point approximation and efficient temporal-difference learning. In *ICML*, pages 43–50, 2001.

[5] A. M. Devraj and S. Meyn. Zap q-learning. In *Advances in Neural Information Processing Systems*, pages 2235–2244, 2017.

[6] A. M. Devraj and S. P. Meyn. Fastest convergence for q-learning. *arXiv preprint arXiv:1707.03770*, 2017.

[7] E. Even-Dar and Y. Mansour. Learning rates for q-learning. *Journal of machine learning Research*, 5(Dec):1–25, 2003.

[8] M. Ghavamzadeh, H. J. Kappen, M. G. Azar, and R. Munos. Speedy q-learning. In *Advances in neural information processing systems*, pages 2411–2419, 2011.

[9] H. Kishner and G. Yin. Stochastic approximation algorithms and applications, 1997.

[10] D.-J. Ma, A. M. Makowski, and A. Shwartz. Stochastic approximations for finite-state markov chains. *Stochastic Processes and Their Applications*, 35(1):27–45, 1990.

[11] S. P. Meyn and R. L. Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.

[12] E. Moulines and F. R. Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems*, pages 451–459, 2011.

[13] D. Paulin et al. Concentration inequalities for markov chains by marton couplings and spectral methods. *Electronic Journal of Probability*, 20, 2015.

[14] C. Szepesvári. The asymptotic convergence-rate of q-learning. In *Advances in Neural Information Processing Systems*, pages 1064–1070, 1998.

[15] J. N. Tsitsiklis. Asynchronous stochastic approximation and q-learning. *Machine learning*, 16 (3):185–202, 1994.

[16] C. J. C. H. Watkins. Learning from delayed rewards. 1989.