Reinforcement Learning

Hidden Theory, and New Super-Fast Algorithms

Tutorial for the Simons Institute program on Real-Time Decision Making

March 7 & 9, 2018

Sean P. Meyn

Based on joint research with Vivek Borkar ... Adithya M. Devraj

COGNITION & CONTROL IN COMPLEX SYSTEMS

Department of Electrical and Computer Engineering — University of Florida

References

[1] V. S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*.

Hindustan Book Agency and Cambridge University Press, Delhi, India and Cambridge, UK, 2008.

[2] A. M. Devraj and S. P. Meyn, *Fastest convergence for Q-learning.*

ArXiv, July 2017.

Tutorial, and extended version of *Zap Q-learning. Advances in Neural Information Processing Systems (NIPS).* Dec. 2017.

More references can be found there, and here: Bibliography

Part I: SA & ML Theory

Survey of basic theory: Borkar's monograph [1] and our tutorial [2]

1. Stochastic Approximation: Algorithm & Motivation
   - Basic Algorithm
   - Monte-Carlo
   - Reinforcement Learning
   - Empirical Risk Minimization
2. ODE Methods
   - Representation in Continuous Time
   - A Menu of ODEs
   - ODE Solidarity: Proof of Convergence
   - SDE Solidarity and Algorithm Performance
3. Optimizing Stochastic Approximation
   - SA for Σ_n
   - Stochastic Newton Raphson

Part II : Zap Q-learning

$$\boxed{\mathsf{E}[f(\theta, W)]\Big|_{\theta=\theta^*} \;=\; 0}$$

Stochastic Approximation

What is Stochastic Approximation?

Why?

A simple goal: Find the solution θ^* to

$$\bar{f}(\theta^*) := \mathsf{E}[f(\theta, W)]\Big|_{\theta=\theta^*} = 0$$

What is Stochastic Approximation?

Why?

A simple goal: Find the solution θ^* to

$$\bar{f}(\theta^*) := \mathsf{E}[f(\theta, W)]\Big|_{\theta=\theta^*} = 0$$

What makes this hard?

What is Stochastic Approximation?

Why?

A simple goal: Find the solution θ^* to

$$\bar{f}(\theta^*) := \mathsf{E}[f(\theta, W)]\Big|_{\theta=\theta^*} = 0$$

What makes this hard?

1. The function f and the distribution of the random vector W may not be known
   – we may only know something about the structure of the problem

What is Stochastic Approximation?

Why?

A simple goal: Find the solution θ^* to

$$\bar{f}(\theta^*) := \mathsf{E}[f(\theta, W)]\Big|_{\theta=\theta^*} = 0$$

What makes this hard?

1. The function f and the distribution of the random vector W may not be known
   – we may only know something about the structure of the problem
2. Even if everything is known, computation of the expectation may be expensive. For root finding, we may need to compute the expectation for many values of θ

What is Stochastic Approximation?

Why?

A simple goal: Find the solution θ^* to

$$\bar{f}(\theta^*) := \mathsf{E}[f(\theta, W)]\Big|_{\theta=\theta^*} = 0$$

What makes this hard?

1. The function f and the distribution of the random vector W may not be known
   – we may only know something about the structure of the problem
2. Even if everything is known, computation of the expectation may be expensive. For root finding, we may need to compute the expectation for many values of θ
3. The recursive algorithms we come up with are often slow, and their variance may be **infinite**: typical in Q-learning [Devraj & M 2017]

What is Stochastic Approximation?

What?

Basic algorithm of Robbins & Monro 1951:

$$\theta(n+1) = \theta(n) + \alpha_n f(\theta(n), W(n+1))$$

What is Stochastic Approximation?

What?

Basic algorithm of Robbins & Monro 1951:

$$\theta(n+1) = \theta(n) + \alpha_n f(\theta(n), W(n+1))$$

The stepsize satisfies

- To ensure we can reach anywhere: $\sum \alpha_n = \infty$
- To attenuate noise: $\sum \alpha_n^2 < \infty$

usually we will take $\alpha_n = 1/n$

What is Stochastic Approximation?

What?

Basic algorithm of Robbins & Monro 1951:

$$\theta(n+1) = \theta(n) + \alpha_n f(\theta(n), W(n+1))$$

The stepsize satisfies

- To ensure we can reach anywhere: $\sum \alpha_n = \infty$
- To attenuate noise: $\sum \alpha_n^2 < \infty$

usually we will take $\alpha_n = 1/n$

Written this way:

$$\theta(n+1) = \theta(n) + \alpha_n[\bar{f}(\theta(n)) + \Delta(n+1)]$$

Interpreted as a noisy Euler approximation to the ODE

$$\frac{d}{dt}x_t = \bar{f}(x_t)$$

Stochastic Approximation Example

Example: Monte-Carlo

Monte-Carlo Estimation

Estimate the mean $\eta = c(X)$, where X is a random variable:

$$\eta = \int c(x)\, f_X(x)\, dx$$

Stochastic Approximation Example

Example: Monte-Carlo

Monte-Carlo Estimation

Estimate the mean $\eta = c(X)$, where X is a random variable

SA interpretation: Find θ^* solving $0 = \mathsf{E}[f(\theta, X)] = \mathsf{E}[c(X) - \theta]$

$$\textit{Algorithm:} \quad \theta(n) = \frac{1}{n}\sum_{i=1}^{n} c(X(i))$$

Stochastic Approximation Example

Example: Monte-Carlo

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

Monte-Carlo Estimation

Estimate the mean $\eta = c(X)$, where X is a random variable

SA interpretation: Find θ^* solving $0 = \mathsf{E}[f(\theta, X)] = \mathsf{E}[c(X) - \theta]$

$$\textit{Algorithm:} \quad \theta(n) = \frac{1}{n}\sum_{i=1}^{n} c(X(i))$$

$$\Longrightarrow \qquad (n+1)\theta(n+1) = \sum_{i=1}^{n+1} c(X(i)) = n\theta(n) + c(X(n+1))$$

$$\Longrightarrow \qquad (n+1)\theta(n+1) = (n+1)\theta(n) + [c(X(n+1)) - \theta(n)]$$

SA Recursion: $\qquad \theta(n+1) = \theta(n) + \alpha_n f(\theta(n), X(n+1))$

SA and RL Design

Functional equations in Stochastic Control

Always of the form

$$0 = \mathsf{E}[F(h^*, \Phi(n+1)) \mid \Phi(0) \ldots \Phi(n)]\,, \qquad h^* = \text{?}$$

SA and RL Design

Functional equations in Stochastic Control

Always of the form

$$0 = \mathsf{E}[F(h^*, \Phi(n+1)) \mid \Phi(0) \ldots \Phi(n)] , \qquad h^* = ?$$

$$\Phi(n) = (\text{state, action})$$

SA and RL Design

Functional equations in Stochastic Control

Always of the form

$$0 = \mathsf{E}[F(h^*, \Phi(n+1)) \mid \Phi(0) \ldots \Phi(n)] \,, \qquad h^* = ?$$

Galerkin relaxation:

$$0 = \mathsf{E}[F(h^{\theta^*}, \Phi(n+1))\zeta_n] \,, \qquad \theta^* = ?$$

SA and RL Design

Functional equations in Stochastic Control

Always of the form

$$0 = \mathsf{E}[F(h^*, \Phi(n+1)) \mid \Phi(0) \ldots \Phi(n)], \qquad h^* = ?$$

Galerkin relaxation:

$$0 = \mathsf{E}[F(h^{\theta^*}, \Phi(n+1))\zeta_n], \qquad \theta^* = ?$$

Necessary Ingredients:

- Parameterized family $\{h^\theta : \theta \in \mathbb{R}^d\}$
- Adapted, d-dimensional stochastic process $\{\zeta_n\}$

SA and RL Design

Functional equations in Stochastic Control

Always of the form

$$0 = \mathsf{E}[F(h^*, \Phi(n+1)) \mid \Phi(0) \dots \Phi(n)]\,, \qquad h^* = ?$$

Galerkin relaxation:

$$0 = \mathsf{E}[F(h^{\theta^*}, \Phi(n+1))\zeta_n]\,, \qquad \theta^* = ?$$

Necessary Ingredients:

- Parameterized family $\{h^\theta : \theta \in \mathbb{R}^d\}$
- Adapted, d-dimensional stochastic process $\{\zeta_n\} \equiv$ *eligibility vectors*

Examples are TD- and Q-Learning

SA and RL Design

Functional equations in Stochastic Control

Always of the form

$$0 = \mathsf{E}[F(h^*, \Phi(n+1)) \mid \Phi(0) \dots \Phi(n)] , \qquad h^* = ?$$

Galerkin relaxation:

$$0 = \mathsf{E}[F(h^{\theta^*}, \Phi(n+1))\zeta_n] , \qquad \theta^* = ?$$

Necessary Ingredients:

- Parameterized family $\{h^\theta : \theta \in \mathbb{R}^d\}$
- Adapted, d-dimensional stochastic process $\{\zeta_n\} \equiv$ *eligibility vectors*

Examples are TD- and Q-Learning

These algorithms are thus special cases of stochastic approximation

Empirical Risk Minimization

Goal: find θ^* that minimizes $J(\theta) = \mathsf{E}[g(\theta, W)]$.

Empirical Risk Minimization

Goal: find θ^* that minimizes $J(\theta) = \mathsf{E}[g(\theta, W)]$.

Settle for empirical risk: $$J_n(\theta) = \frac{1}{n}\sum_{k=1}^{n} g(\theta, W_k)$$

Methods to compute minimizer θ^*_n quickly

focus of current research – e.g., [14].

Empirical Risk Minimization

Goal: find θ^* that minimizes $J(\theta) = \mathsf{E}[g(\theta, W)]$.

Settle for empirical risk: $$J_n(\theta) = \frac{1}{n}\sum_{k=1}^{n} g(\theta, W_k)$$

Methods to compute minimizer θ_n^* quickly

focus of current research – e.g., [14].

However, don't forget the original problem:

$$\theta_n^* - \theta^* \overset{\text{dist}}{\approx} \frac{1}{\sqrt{n}} N(0, \Sigma^*)$$

Formula for covariance below

Empirical Risk Minimization

Goal: find θ^* that minimizes $J(\theta) = \mathsf{E}[g(\theta, W)]$.

Settle for empirical risk: $$J_n(\theta) = \frac{1}{n}\sum_{k=1}^{n} g(\theta, W_k)$$

Methods to compute minimizer θ_n^* quickly

focus of current research – e.g., [14].

However, don't forget the original problem:

$$\theta_n^* - \theta^* \stackrel{\mathrm{dist}}{\approx} \frac{1}{\sqrt{n}} N(0, \Sigma^*)$$

Formula for covariance below

The same conclusion would be reached using stochastic approximation (with careful design).

ODE and SDE Approximations

Continuous time interpolation

The starting point of all approximations:

1. Timescale: $t_0 = 0$ and $t_{n+1} = t_n + \alpha_n$ for $n \geq 0$.
2. Continuous time process: $X_t = \theta(n)$ for $t = t_n$; defined elsewhere by linear interpolation.

ODE and SDE Approximations

Continuous time interpolation

The starting point of all approximations:

1. Timescale: $t_0 = 0$ and $t_{n+1} = t_n + \alpha_n$ for $n \geq 0$.
2. Continuous time process: $X_t = \theta(n)$ for $t = t_n$; defined elsewhere by linear interpolation.

For $t_n > t_k$,

$$X_{t_n} = X_{t_k} + \sum_j f(X_{t_j}, W(j+1))\,\delta_{t_j}\,, \qquad \delta_{t_j} = t_j - t_{j-1}$$

$$= X_{t_k} + \int_{t_k}^{t_n} \bar{f}(X_s)\,ds + \mathcal{E}(t_k, t_n)$$

ODE and SDE Approximations

Continuous time interpolation

The starting point of all approximations:

1. Timescale: $t_0 = 0$ and $t_{n+1} = t_n + \alpha_n$ for $n \geq 0$.
2. Continuous time process: $X_t = \theta(n)$ for $t = t_n$; defined elsewhere by linear interpolation.

For $t_n > t_k$,

$$\begin{aligned} X_{t_n} &= X_{t_k} + \sum_j f(X_{t_j}, W(j+1))\, \delta_{t_j}\,, \qquad \delta_{t_j} = t_j - t_{j-1} \\ &= X_{t_k} + \int_{t_k}^{t_n} \bar{f}(X_s)\, ds + \mathcal{E}(t_k, t_n) \end{aligned}$$

Properties of the noise follow from assumptions on f and $\boldsymbol{W}$.

ODE and SDE Approximations

Continuous time interpolation

The starting point of all approximations:

1. Timescale: $t_0 = 0$ and $t_{n+1} = t_n + \alpha_n$ for $n \geq 0$.
2. Continuous time process: $X_t = \theta(n)$ for $t = t_n$; defined elsewhere by linear interpolation.
3. Time horizon $T \gg 0$: Construct increasing subsequence $\{T_n\}$ so that

$$T = \lim_{n\to\infty} (T_{n+1} - T_n)$$

Analysis restricted to each time interval:

$$X_t = X_{T_n} + \int_{T_n}^{t} \bar{f}(X_s)\, ds + \mathcal{E}(T_n, t)\,, \quad T_n \leq t < T_{n+1}$$

ODE and SDE Approximations

Properties of the noise follow from assumptions on f and $\boldsymbol{W}$.

Continuous time process: $X_t = \theta(n)$ for $t = t_n$:

$$X_t = X_{T_n} + \int_{T_n}^{t} \bar{f}(X_s)\, ds + \mathcal{E}(T_n, t)\,, \quad T_n \le t < T_{n+1}$$

ODE and SDE Approximations

Properties of the noise follow from assumptions on f and $\boldsymbol{W}$.

Continuous time process: $X_t = \theta(n)$ for $t = t_n$:

$$X_t = X_{T_n} + \int_{T_n}^{t} \bar{f}(X_s)\,ds + \mathcal{E}(T_n, t)\,, \quad T_n \le t < T_{n+1}$$

For $\alpha_k = k^{-1}$,

$$\mathcal{E}(t_m, t_n) = \sum_{k=m+1}^{n} \big[f(\theta(k), W(k+1)) - \bar{f}(\theta(k))\big]\alpha_k + O(m^{-2})$$

ODE and SDE Approximations

Properties of the noise follow from assumptions on f and $\boldsymbol{W}$.

Continuous time process: $X_t = \theta(n)$ for $t = t_n$:

$$X_t = X_{T_n} + \int_{T_n}^{t} \bar{f}(X_s)\, ds + \mathcal{E}(T_n, t)\,, \quad T_n \le t < T_{n+1}$$

For $\alpha_k = k^{-1}$,

$$\mathcal{E}(t_m, t_n) = \sum_{k=m+1}^{n} \big[f(\theta(k), W(k+1)) - \bar{f}(\theta(k)) \big] \alpha_k + O(m^{-2})$$

For nice Markovian $\boldsymbol{W}$, f Lipschitz in θ and "nice" in W:

$$\mathcal{E}(t_m, t_n) = M(t_n) - M(t_m) + \mathcal{J}(t_m, t_n)$$

where $\boldsymbol{M}$ is a martingale, and the "junk term" can be disposed of.

ODE and SDE Approximations !! Comments for the experts

Properties of the noise follow from assumptions on f and $\boldsymbol{W}$.

Continuous time process: $X_t = \theta(n)$ for $t = t_n$:

$$X_t = X_{T_n} + \int_{T_n}^{t} \bar{f}(X_s)\,ds + \mathcal{E}(T_n, t)\,, \quad T_n \le t < T_{n+1}$$

$$\begin{aligned}\mathcal{E}(t_m, t_n) &= \sum_{k=m+1}^{n} \big[f(\theta(k), W(k+1)) - \bar{f}(\theta(k))\big]\alpha_k + O(m^{-2}) \\ &= M(t_n) - M(t_m) + \mathcal{J}(t_m, t_n)\end{aligned}$$

Markovian $\boldsymbol{W}$: *what is nice?*

$$\mathcal{J}(t_m, t_n) = \text{Simple junk} - \sum_{k=m+1}^{n} \alpha_k[\mathcal{H}_k - \mathcal{H}_{k-1}]$$

Need nice solutions to "Poisson's equation": $\mathcal{H}_k = h(\theta(k), W(k+1))$ [6, 7].

ODE and SDE Approximations

- Boundedness of $\{\theta_n\}$ [1, Ch. 3]
  Follows from stability of the homogeneous ODE,

$$\frac{d}{dt}\xi_t = \bar{f}^{\infty}(\xi_t)\,, \qquad \bar{f}^{\infty}(x) = \lim_{r\to\infty} r^{-1}\bar{f}(rx)$$ Borkar-M. Theorem

"*ODE at* ∞"

ODE and SDE Approximations

- Boundedness of $\{\theta_n\}$ [1, Ch. 3]
  Follows from stability of the homogeneous ODE,

$$\frac{d}{dt}\xi_t = \bar{f}^{\infty}(\xi_t)\,, \qquad \bar{f}^{\infty}(x) = \lim_{r\to\infty} r^{-1}\bar{f}(rx)$$

Borkar-M. Theorem

- Convergence of $\{\theta_n\}$ to θ^* [1, Ch. 2]
  $X_t \approx x_t^k$ for large k and all t, where

$$\frac{d}{dt}x_t^k = \bar{f}(x_t^k)\,, \quad x_{T_k}^k = X_{T_k}$$

ODE and SDE Approximations

- Boundedness of $\{\theta_n\}$ [1, Ch. 3]
  Follows from stability of the homogeneous ODE,

$$\frac{d}{dt}\xi_t = \bar{f}^\infty(\xi_t)\,, \qquad \bar{f}^\infty(x) = \lim_{r\to\infty} r^{-1}\bar{f}(rx)$$

Borkar-M. Theorem

- Convergence of $\{\theta_n\}$ to θ^* [1, Ch. 2]
  $X_t \approx x_t^k$ for large k and all t, where

$$\frac{d}{dt}x_t^k = \bar{f}(x_t^k)\,, \quad x_{T_k}^k = X_{T_k}$$

- Variance analysis $\equiv$ SDE approximation [1, Ch. 8]

$$Y_T \approx Y_0 + \int_0^T (A + \tfrac{1}{2}I)Y_s\,ds + B_T$$

$Y_t = e^{t/2}(X_t - \theta^*) \qquad Y_{t_n} \approx \sqrt{n}(\theta(n) - \theta^*)$ since $t_n \approx \log(n)$.

Algorithm and Convergence Analysis

Convergence of $\{\theta_n\}$ to θ^* In one word: Euler scheme for solving an ODE is robust

Comparison

$$X_t = X_{T_n} + \int_{T_n}^{t} \bar{f}(X_s)\, ds + \mathcal{E}(T_n, t)$$

$$x_t^n = X_{T_n} + \int_{T_n}^{t} \bar{f}(x_s^n)\, ds\,, \qquad T_n \le t < T_{n+1}$$

Algorithm and Convergence Analysis

Convergence of $\{\theta_n\}$ to θ^* In one word: Euler scheme for solving an ODE is robust

Comparison

$$X_t = X_{T_n} + \int_{T_n}^t \bar{f}(X_s)\, ds + \mathcal{E}(T_n, t)$$

$$x_t^n = X_{T_n} + \int_{T_n}^t \bar{f}(x_s^n)\, ds\,, \qquad T_n \le t < T_{n+1}$$

Assumptions

- $\frac{d}{dt} x_t = \bar{f}(x_t)$ is globally asymptotically stable
- $\bar{f}$ is Lipschiz continuous, Lipschitz constant L

Algorithm and Convergence Analysis

Convergence of $\{\theta_n\}$ to θ^*

Comparison

$$X_t = X_{T_n} + \int_{T_n}^t \bar{f}(X_s)\, ds + \mathcal{E}(T_n, t)$$
$$x_t^n = X_{T_n} + \int_{T_n}^t \bar{f}(x_s^n)\, ds\,, \qquad T_n \le t < T_{n+1}$$

Assumptions

- $\frac{d}{dt}x_t = \bar{f}(x_t)$ is globally asymptotically stable
- $\bar{f}$ is Lipschiz continuous, Lipschitz constant L
- Nice noise: $\lim_{n\to\infty} \max_{T_n \le t \le T_{n+1}} \|\mathcal{E}(T_n, t)\| = 0.$
- The sequence $\{\theta_n\}$ is bounded (Lyapunov condition, or check *ODE at* ∞)

Algorithm and Convergence Analysis

Convergence of $\{\theta_n\}$ to θ^*

Comparison

$$X_t = X_{T_n} + \int_{T_n}^{t} \bar{f}(X_s)\, ds + \mathcal{E}(T_n, t)$$

$$x_t^n = X_{T_n} + \int_{T_n}^{t} \bar{f}(x_s^n)\, ds\,, \qquad T_n \le t < T_{n+1}$$

Error: $e_t^n = \|X_t - x_t^n\|$ and $\bar{\mathcal{E}}^n = \max_{T_n \le t \le T_{n+1}} \|\mathcal{E}(T_n, t)\|$:

Algorithm and Convergence Analysis

Convergence of $\{\theta_n\}$ to θ^*

Comparison

$$X_t = X_{T_n} + \int_{T_n}^{t} \bar{f}(X_s)\, ds + \mathcal{E}(T_n, t)$$

$$x_t^n = X_{T_n} + \int_{T_n}^{t} \bar{f}(x_s^n)\, ds\,, \qquad T_n \le t < T_{n+1}$$

Error: $e_t^n = \|X_t - x_t^n\|$ and $\bar{\mathcal{E}}^n = \max_{T_n \le t \le T_{n+1}} \|\mathcal{E}(T_n, t)\|$:

$$e_t^n \le L \int_{T_n}^{t} e_s^n\, ds + \bar{\mathcal{E}}^n\,, \qquad T_n \le t < T_{n+1}$$

Algorithm and Convergence Analysis

Convergence of $\{\theta_n\}$ to θ^*

Comparison

$$X_t = X_{T_n} + \int_{T_n}^t \bar{f}(X_s)\,ds + \mathcal{E}(T_n, t)$$

$$x_t^n = X_{T_n} + \int_{T_n}^t \bar{f}(x_s^n)\,ds\,, \qquad T_n \le t < T_{n+1}$$

Error: $e_t^n = \|X_t - x_t^n\|$ and $\bar{\mathcal{E}}^n = \max_{T_n \le t \le T_{n+1}} \|\mathcal{E}(T_n, t)\|$:

$$e_t^n \le L \int_{T_n}^t e_s^n\,ds + \bar{\mathcal{E}}^n\,, \qquad T_n \le t < T_{n+1}$$

$$\Longrightarrow \quad e_t^n \le \bar{\mathcal{E}}^n \exp([T_{n+1} - T_n]L) \qquad \text{Bellman Gronwall Lemma}$$

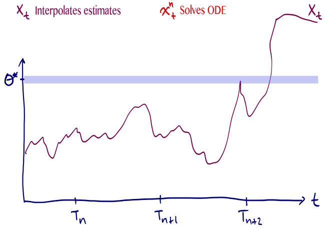
Algorithm and Convergence Analysis

Convergence of $\{\theta_n\}$ to θ^*

Error: $e_t^n = \|X_t - x_t^n\|$ and $\bar{\mathcal{E}}^n = \max_{T_n \le t \le T_{n+1}} \|\mathcal{E}(T_n, t)\|$:

$$e_t^n \le \bar{\mathcal{E}}^n \exp([T_{n+1} - T_n]L) \qquad \textit{vanishes}$$

$$\Longrightarrow \quad X_t \approx x_t^n \quad \textit{for large } n \textit{ and all } t$$



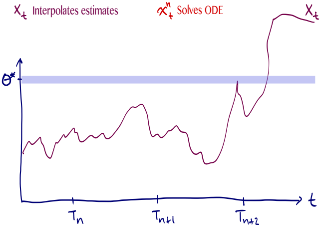
Algorithm and Convergence Analysis

Convergence of $\{\theta_n\}$ to θ^*

Error: $e^n_t = \|X_t - x^n_t\|$ and $\bar{\mathcal{E}}^n = \max_{T_n \le t \le T_{n+1}} \|\mathcal{E}(T_n, t)\|$:

$$e^n_t \le \bar{\mathcal{E}}^n \exp([T_{n+1} - T_n]L) \qquad \textit{vanishes}$$

$$\Longrightarrow \quad X_t \approx x^n_t \quad \textit{for large } n \textit{ and all } t$$



Fix large T, and note implications:

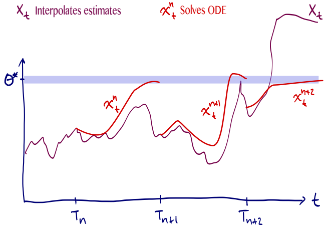
Algorithm and Convergence Analysis

Convergence of $\{\theta_n\}$ to θ^*

Error: $e_t^n = \|X_t - x_t^n\|$ and $\bar{\mathcal{E}}^n = \max_{T_n \le t \le T_{n+1}} \|\mathcal{E}(T_n, t)\|$:

$$e_t^n \le \bar{\mathcal{E}}^n \exp([T_{n+1} - T_n]L) \qquad \textit{vanishes}$$

$$\implies \quad X_t \approx x_t^n \quad \textit{for large } n \textit{ and all } t$$



Fix large T, and note implications:

1. $x^n_{T_{n+1}-} \approx \theta^*$ *for all* n by GAS

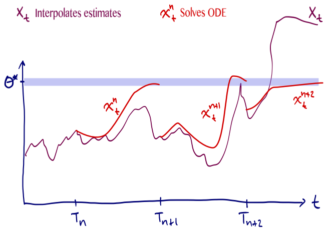
Algorithm and Convergence Analysis

Convergence of $\{\theta_n\}$ to θ^*

Error: $e^n_t = \|X_t - x^n_t\|$ and $\bar{\mathcal{E}}^n = \max_{T_n \le t \le T_{n+1}} \|\mathcal{E}(T_n, t)\|$:

$$e^n_t \le \bar{\mathcal{E}}^n \exp([T_{n+1} - T_n]L) \qquad \textit{vanishes}$$

$$\Longrightarrow \quad X_t \approx x^n_t \quad \textit{for large } n \textit{ and all } t$$



Fix large T, and note implications:

1. $x^n_{T_{n+1}-} \approx \theta^*$ *for all* n by GAS
2. $x^{n+1}_{T_{n+1}} = X_{T_{n+1}} \approx x^n_{T_{n+1}-} \approx \theta^*$ *for large* n

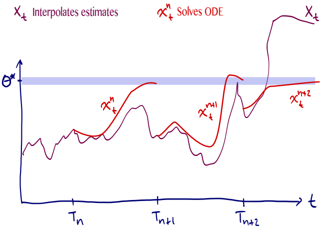
Algorithm and Convergence Analysis

Convergence of $\{\theta_n\}$ to θ^*

Error: $e^n_t = \|X_t - x^n_t\|$ and $\bar{\mathcal{E}}^n = \max_{T_n \le t \le T_{n+1}} \|\mathcal{E}(T_n, t)\|$:

$$e^n_t \le \bar{\mathcal{E}}^n \exp([T_{n+1} - T_n]L) \qquad \textit{vanishes}$$

$$\Longrightarrow \quad X_t \approx x^n_t \quad \textit{for large } n \textit{ and all } t$$



Fix large T, and note implications:

1. $x^n_{T_{n+1}-} \approx \theta^*$ *for all* n by GAS
2. $x^{n+1}_{T_{n+1}} = X_{T_{n+1}} \approx x^n_{T_{n+1}-} \approx \theta^*$ *for large* n
3. $x^{n+1}_t \approx \theta^*$ *for large* n *and all* t by GAS

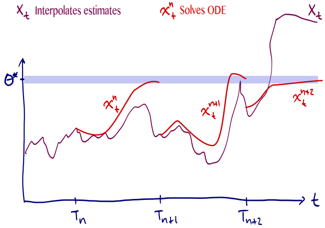
Algorithm and Convergence Analysis

Convergence of $\{\theta_n\}$ to θ^*

Error: $e_t^n = \|X_t - x_t^n\|$ and $\bar{\mathcal{E}}^n = \max_{T_n \le t \le T_{n+1}} \|\mathcal{E}(T_n, t)\|$:

$$e_t^n \le \bar{\mathcal{E}}^n \exp([T_{n+1} - T_n]L) \qquad \textit{vanishes}$$

$$\Longrightarrow \quad X_t \approx x_t^n \quad \textit{for large } n \textit{ and all } t$$



Fix large T, and note implications:

1. $x^n_{T_{n+1}-} \approx \theta^*$ *for all* n by GAS
2. $x^{n+1}_{T_{n+1}} = X_{T_{n+1}} \approx x^n_{T_{n+1}-} \approx \theta^*$ *for large* n
3. $x_t^{n+1} \approx \theta^*$ *for large* n *and all* t by GAS
4. $X_t \approx x_t^n \approx \theta^*$ *for large* n *and all* t

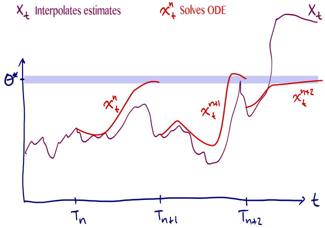
Algorithm and Convergence Analysis

Convergence of $\{\theta_n\}$ to θ^*

Error: $e_t^n = \|X_t - x_t^n\|$ and $\bar{\mathcal{E}}^n = \max_{T_n \le t \le T_{n+1}} \|\mathcal{E}(T_n, t)\|$:

$$e_t^n \le \bar{\mathcal{E}}^n \exp([T_{n+1} - T_n]L) \qquad \textit{vanishes}$$

$$\Longrightarrow \quad X_t \approx x_t^n \quad \textit{for large } n \textit{ and all } t$$



Fix large T, and note implications:

1. $x^n_{T_{n+1}-} \approx \theta^*$ *for all* n by GAS
2. $x^{n+1}_{T_{n+1}} = X_{T_{n+1}} \approx x^n_{T_{n+1}-} \approx \theta^*$ *for large* n
3. $x^{n+1}_t \approx \theta^*$ *for large* n *and all* t by GAS
4. $X_t \approx x^n_t \approx \theta^*$ *for large* n *and all* t

Convergence: $\lim_{k\to\infty} \theta_k = \lim_{k\to\infty} X_{t_k} = \theta^*$

SDE Approximations

Linear SDE for $Y_t = e^{t/2}(X_t - \theta^*)$

$$Y_{t_n} \approx \sqrt{n}(\theta(n) - \theta^*) \text{ since } t_n \approx \log(n).$$

- Same starting point: $X_t = X_{T_n} + \int_{T_n}^{t} \bar{f}(X_s)\, ds + \mathcal{E}(T_n, t)$
- Linearize: $\bar{f}(x) \approx A(x - \theta^*)$, for $x \approx \theta^*$.
- Nice noise gives FCLT: $e^{(t-T_n)/2}\mathcal{E}(T_n, t) \overset{\text{dist}}{\approx} B_t - B_{T_n}$

SDE Approximations

Linear SDE for $Y_t = e^{t/2}(X_t - \theta^*)$

$$Y_{t_n} \approx \sqrt{n}(\theta(n) - \theta^*) \text{ since } t_n \approx \log(n).$$

- Same starting point: $X_t = X_{T_n} + \int_{T_n}^{t} \bar{f}(X_s)\,ds + \mathcal{E}(T_n, t)$
- Linearize: $\bar{f}(x) \approx A(x - \theta^*)$, for $x \approx \theta^*$.
- Nice noise gives FCLT: $e^{(t-T_n)/2}\mathcal{E}(T_n, t) \stackrel{\text{dist}}{\approx} B_t - B_{T_n}$

and with a bit of work:

$$Y_t \stackrel{\text{dist}}{\approx} Y_{T_n} + \int_{T_n}^{t} (A + \tfrac{1}{2}I) Y_s\,ds + B_t - B_{T_n}$$

SDE Approximations

Linear SDE for $Y_t = e^{t/2}(X_t - \theta^*)$

$$Y_t \overset{\text{dist}}{\approx} Y_{T_n} + \int_{T_n}^{t} (A + \tfrac{1}{2}I) Y_s \, ds + B_t - B_{T_n}$$

$\boldsymbol{B}$ Brownian motion, $B_t \sim N(0, t\Sigma_\Delta)$.

Translating back to reality: (under assumptions I won't list)

SDE Approximations

Linear SDE for $Y_t = e^{t/2}(X_t - \theta^*)$

$$Y_t \overset{\text{dist}}{\approx} Y_{T_n} + \int_{T_n}^{t} (A + \tfrac{1}{2}I)Y_s\, ds + B_t - B_{T_n}$$

$\boldsymbol{B}$ Brownian motion, $B_t \sim N(0, t\Sigma_\Delta)$.

Translating back to reality: (under assumptions I won't list)

Central Limit Theorem

$\sqrt{n}\tilde{\theta}(n)$ converges in distribution to $N(0, \Sigma)$, whose covariance is the solution to the Lyapunov equation:

$$(A + \tfrac{1}{2}I)\Sigma + \Sigma(A + \tfrac{1}{2}I)^{\intercal} + \Sigma_\Delta = 0$$

The covariance is finite if Real $\lambda(A) < -\frac{1}{2}$

SDE Approximations

Linear SDE for $Y_t = e^{t/2}(X_t - \theta^*)$

Central Limit Theorem

$\sqrt{n}\tilde{\theta}(n)$ converges in distribution to $N(0, \Sigma)$, whose covariance is the solution to the Lyapunov equation:

$$(A + \tfrac{1}{2}I)\Sigma + \Sigma(A + \tfrac{1}{2}I)^{\intercal} + \Sigma_{\Delta} = 0$$

The covariance is finite if Real $\lambda(A) < -\frac{1}{2}$

Questions for algorithm design:

1. How do we fix an algorithm if it fails this condition?
2. How can we optimize Σ?

SDE Approximations

Linear SDE for $Y_t = e^{t/2}(X_t - \theta^*)$

Central Limit Theorem

$\sqrt{n}\tilde{\theta}(n)$ converges in distribution to $N(0, \Sigma)$, whose covariance is the solution to the Lyapunov equation:

$$(A + \tfrac{1}{2}I)\Sigma + \Sigma(A + \tfrac{1}{2}I)^{\intercal} + \Sigma_\Delta = 0$$

The covariance is finite if Real $\lambda(A) < -\frac{1}{2}$

Questions for algorithm design:

1. How do we fix an algorithm if it fails this condition?
2. How can we optimize Σ?
3. Does this lead to improved algorithms for reinforcement learning?

Asymptotic Covariance

Recursion for uncorrelated noise

Consider a linear model with $\tilde{\theta}(n) := \theta(n) - \theta^*$:

$$\tilde{\theta}(n+1) = \tilde{\theta}(n) + \tfrac{1}{n}[A\tilde{\theta}(n) + \Delta(n+1)]$$

$\{\Delta(n)\}$ uncorrelated, zero mean, covariance Σ_Δ.

Asymptotic Covariance

Recursion for uncorrelated noise

Consider a linear model with $\tilde{\theta}(n) := \theta(n) - \theta^*$:

$$\tilde{\theta}(n+1) = \tilde{\theta}(n) + \tfrac{1}{n}[A\tilde{\theta}(n) + \Delta(n+1)]$$

$\{\Delta(n)\}$ uncorrelated, zero mean, covariance Σ_Δ.

Approximate $\sqrt{n+1} \approx \sqrt{n}(1 + (2n)^{-1})$:

$$\sqrt{n+1}\tilde{\theta}(n+1) \approx \sqrt{n}\tilde{\theta}(n) + \tfrac{1}{n}[(A + \tfrac{1}{2}I)\sqrt{n}\tilde{\theta}(n) + \sqrt{n}\Delta(n+1)]$$

Asymptotic Covariance

Recursion for uncorrelated noise

Consider a linear model with $\tilde{\theta}(n) := \theta(n) - \theta^*$:

$$\tilde{\theta}(n+1) = \tilde{\theta}(n) + \tfrac{1}{n}[A\tilde{\theta}(n) + \Delta(n+1)]$$

$\{\Delta(n)\}$ uncorrelated, zero mean, covariance Σ_Δ.

Approximate $\sqrt{n+1} \approx \sqrt{n}(1 + (2n)^{-1})$:

$$\sqrt{n+1}\tilde{\theta}(n+1) \approx \sqrt{n}\tilde{\theta}(n) + \tfrac{1}{n}[(A + \tfrac{1}{2}I)\sqrt{n}\tilde{\theta}(n) + \sqrt{n}\Delta(n+1)]$$

Covariance recursion:

$$\begin{aligned}\Sigma_{n+1} &= (n+1)\mathsf{E}\Big[\tilde{\theta}(n+1)\tilde{\theta}(n+1)^{\intercal}\Big] \\ &\approx \Sigma_n + \tfrac{1}{n}\Big\{(A + \tfrac{1}{2}I)\Sigma_n + \Sigma_n(A + \tfrac{1}{2}I)^{\intercal} + \Sigma_\Delta\Big\}\end{aligned}$$

Asymptotic Covariance

$$\Sigma = \lim_{n\to\infty} \Sigma_n = \lim_{n\to\infty} n\mathsf{E}\big[\tilde\theta(n)\tilde\theta(n)^{\intercal}\big]\,, \qquad \sqrt{n}\tilde\theta(n) \approx N(0,\Sigma)$$

SA recursion for covariance:

$$\Sigma_{n+1} \approx \Sigma_n + \frac{1}{n}\Big\{(A+\tfrac{1}{2}I)\Sigma_n + \Sigma_n(A+\tfrac{1}{2}I)^{\intercal} + \Sigma_\Delta\Big\}$$

$$A = \frac{d}{d\theta}\bar f\,(\theta^*)$$

Conclusions

1. If Re $\lambda(A) \ge -\frac{1}{2}$ for some eigenvalue then Σ is (typically) infinite
2. If Re $\lambda(A) < -\frac{1}{2}$ for all, then $\Sigma = \lim_{n\to\infty}\Sigma_n$ is the unique solution to the Lyapunov equation:

$$\boxed{0 = (A+\tfrac{1}{2}I)\Sigma + \Sigma(A+\tfrac{1}{2}I)^{\intercal} + \Sigma_\Delta}$$

Optimal Asymptotic Covariance

Introduce a $d \times d$ matrix gain sequence $\{G_n\}$:

$$\theta(n+1) = \theta(n) + \frac{1}{n+1} G_n f(\theta(n), X(n))$$

Optimal Asymptotic Covariance

Introduce a $d \times d$ matrix gain sequence $\{G_n\}$:

$$\theta(n+1) = \theta(n) + \frac{1}{n+1} G_n f(\theta(n), X(n))$$

Assume it converges, and linearize:

$$\tilde{\theta}(n+1) \approx \tilde{\theta}(n) + \frac{1}{n+1} G\big(A\tilde{\theta}(n) + \Delta(n+1)\big), \qquad A = \frac{d}{d\theta}\bar{f}\,(\theta^*)\,.$$

Optimal Asymptotic Covariance

Introduce a $d \times d$ matrix gain sequence $\{G_n\}$:

$$\theta(n+1) = \theta(n) + \frac{1}{n+1} G_n f(\theta(n), X(n))$$

Assume it converges, and linearize:

$$\tilde{\theta}(n+1) \approx \tilde{\theta}(n) + \frac{1}{n+1} G\big(A\tilde{\theta}(n) + \Delta(n+1)\big), \qquad A = \frac{d}{d\theta}\bar{f}\,(\theta^*)\,.$$

If $G = G^* := -A^{-1}$ then

- Resembles Monte-Carlo estimate
- Resembles Newton-Rapshon *Stochastic Gauss-Newton*, Ruppert [9]
- It is optimal: $\Sigma^* = G^* \Sigma_\Delta G^{*\,T} \leq \Sigma^G$ *any other* G

Optimal Asymptotic Covariance

Introduce a $d \times d$ matrix gain sequence $\{G_n\}$:

$$\theta(n+1) = \theta(n) + \frac{1}{n+1} G_n f(\theta(n), X(n))$$

Assume it converges, and linearize:

$$\tilde{\theta}(n+1) \approx \tilde{\theta}(n) + \frac{1}{n+1} G\big(A\tilde{\theta}(n) + \Delta(n+1)\big), \qquad A = \frac{d}{d\theta}\bar{f}\,(\theta^*)\,.$$

If $G = G^* := -A^{-1}$ then

- Resembles Monte-Carlo estimate
- Resembles Newton-Rapshon   *Stochastic Gauss-Newton*, Ruppert [9]
- It is optimal: $\Sigma^* = G^* \Sigma_\Delta {G^*}^T \leq \Sigma^G$ *any other* G

Ruppert-Polyak averaging is also optimal, but first two bullets are missing.

Optimal Asymptotic Covariance

Example: return to Monte-Carlo

$$\theta(n+1) = \theta(n) + \frac{g}{n+1}\Big(-\theta(n) + X(n+1)\Big)$$

Optimal Asymptotic Covariance

Example: return to Monte-Carlo

$$\theta(n+1) = \theta(n) + \frac{g}{n+1}\Big(-\theta(n) + X(n+1)\Big)$$

$$\Delta(n) = X(n) - \mathsf{E}[X(n)]$$

Optimal Asymptotic Covariance

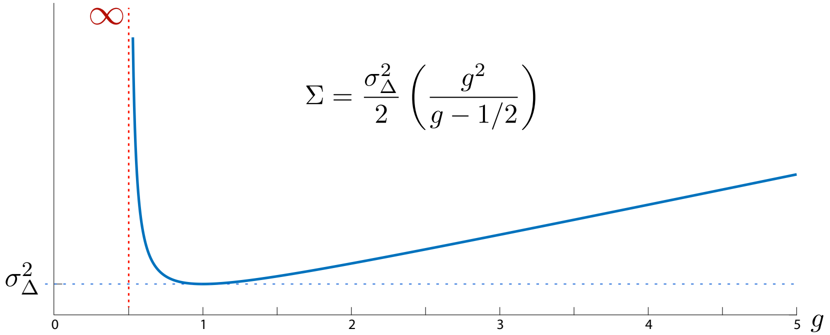
Normalization for analysis: $\Delta(n) = X(n) - \mathsf{E}[X(n)]$

$$\tilde{\theta}(n+1) = \tilde{\theta}(n) + \frac{g}{n+1}\Big(-\tilde{\theta}(n) + \Delta(n+1)\Big)$$

Optimal Asymptotic Covariance

Normalization for analysis: $\Delta(n) = X(n) - \mathsf{E}[X(n)]$

$$\tilde{\theta}(n+1) = \tilde{\theta}(n) + \frac{g}{n+1}\Big(-\tilde{\theta}(n) + \Delta(n+1)\Big)$$



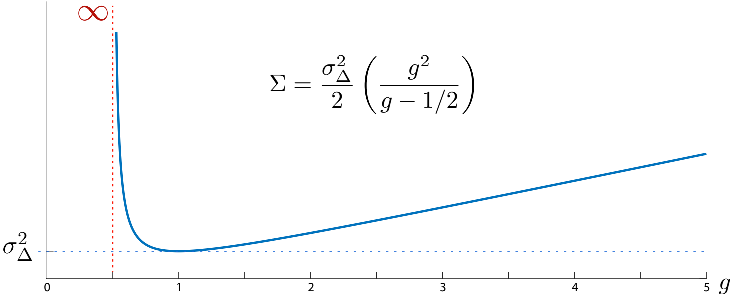
Asymptotic variance as a function of g

Optimal Asymptotic Covariance

Normalization for analysis: $\Delta(n) = X(n) - \mathsf{E}[X(n)]$

$$\tilde{\theta}(n+1) = \tilde{\theta}(n) + \frac{g}{n+1}\Big(-\tilde{\theta}(n) + \Delta(n+1)\Big)$$

Example: $X(n) = W^2(n)$, $W \sim N(0,1)$, $\sigma^2_\Delta = 2$



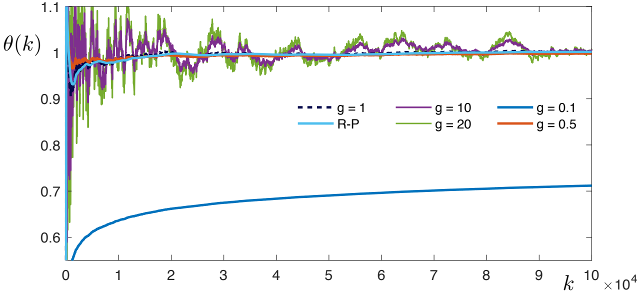
Asymptotic variance as a function of g

Optimal Asymptotic Covariance

Normalization for analysis: $\Delta(n) = X(n) - \mathsf{E}[X(n)]$

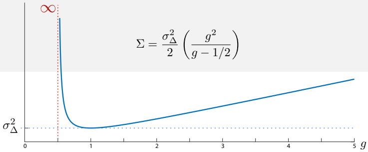
$$\tilde{\theta}(n+1) = \tilde{\theta}(n) + \frac{g}{n+1}\Big(-\tilde{\theta}(n) + \Delta(n+1)\Big)$$

Example: $X(n) = W^2(n)$, $W \sim N(0,1)$, $\sigma^2_\Delta = 2$

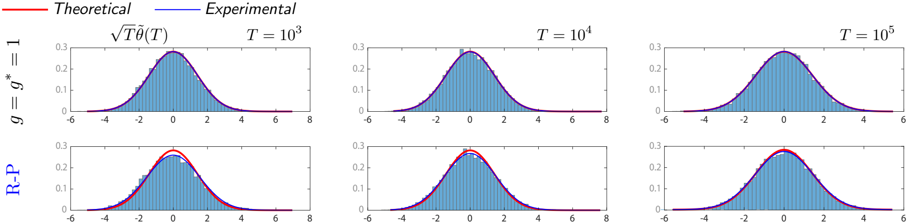


SA estimates of $\mathsf{E}[W^2]$, $W \sim N(0,1)$

Optimal Asymptotic Covariance



Central Limit Theorem optimal $g^* = 1$



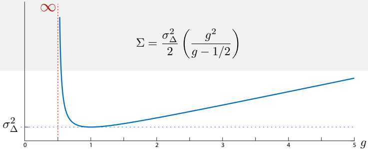
Ruppert-Polyak: *turn up the gain, with* $\varrho \in (0.5, 1)$*:*

$$\bar{\theta}(n+1) = \bar{\theta}(n) + \frac{1}{(n+1)^{\varrho}}\left[-\bar{\theta}(n) + X(n+1)\right]$$

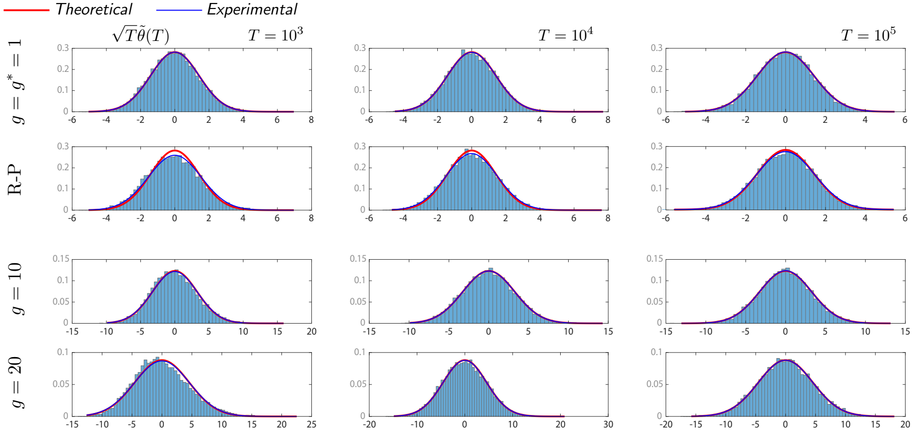
$$\theta(n) = \frac{1}{n}\sum_{k=1}^{n}\bar{\theta}(k)$$

Also has optimal asymptotic covariance

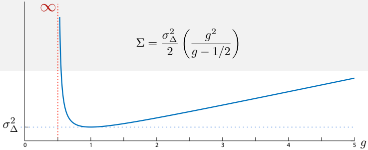
Optimal Asymptotic Covariance



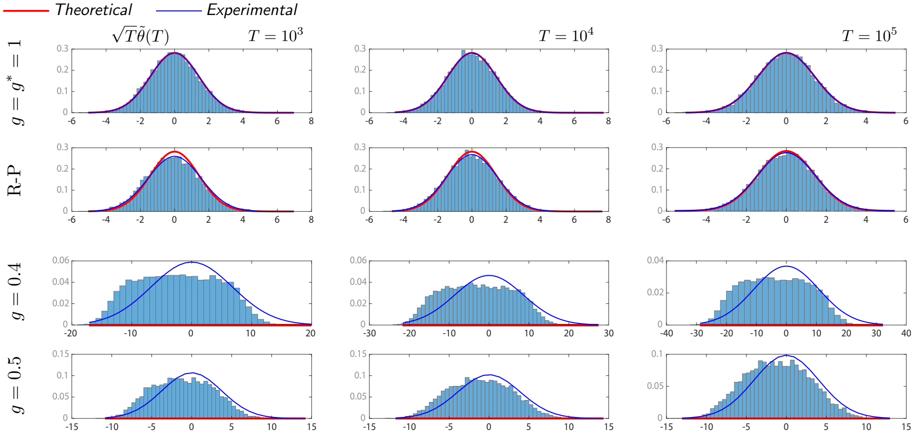
Central Limit Theorem sub-optimal $g > 1$



Optimal Asymptotic Covariance

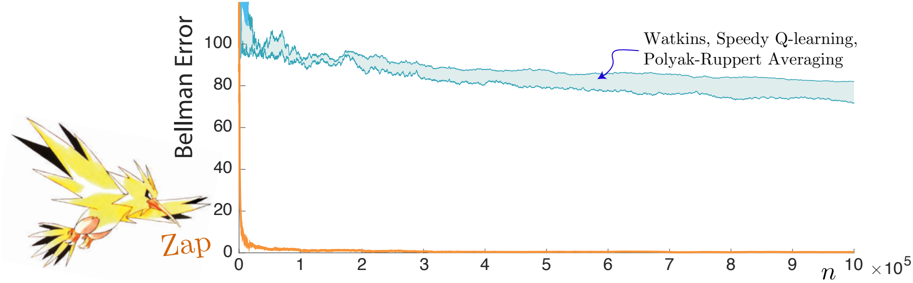


Central Limit Theorem fails $g \leq 1/2$



Optimal Asymptotic Covariance

Impact on algorithm design : new Q-learning algorithms

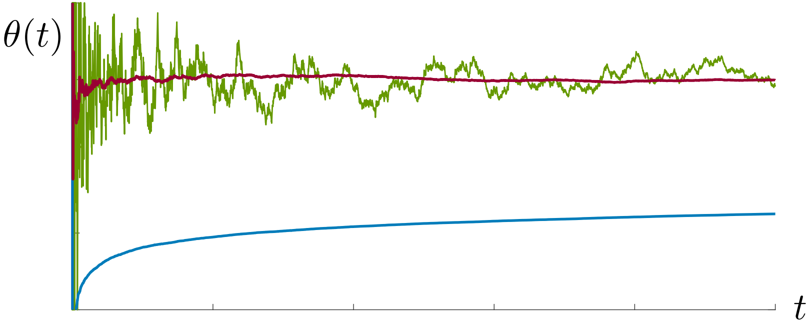


Next time

Part II: Fastest SA and Zap Q-Learning

Hidden theory implications for reinforcement learning

4. Fastest Stochastic Approximation
   - Algorithm Performance Revisited
   - Zap Stochastic Newton-Raphson
5. Reinforcement Learning
   - RL & SA
   - MDP Theory
   - Q-Learning
6. Zap Q-Learning
   - Watkin's algorithm
   - Optimal stopping
7. Conclusions & Future Work
8. References



Fastest Stochastic Approximation

What is Stochastic Approximation?

Recap

Basic algorithm of Robbins & Monro 1951, with matrix gain:

$$\theta(n+1) = \theta(n) + \alpha_n G_n f(\theta(n), W(n+1))$$

Interpreted as a noisy Euler approximation to the ODE

$$\frac{d}{dt} x_t = G\bar{f}(x_t)$$

Usually we take $\alpha_n = 1/n$
Matrices $\{G_n\}$ used to

- Optimize asymptotic covariance
- Improve dynamics (inspired by Newton-Raphson)

Performance Criteria

Two standard approaches to evaluate performance, $\tilde{\theta}(n) := \theta(n) - \theta^*$:

1. Finite-n bound:

$$\mathsf{P}\{\|\tilde{\theta}(n)\| \geq \varepsilon\} \leq \exp(-I(\varepsilon, n)), \qquad I(\varepsilon, n) = O(n\varepsilon^2)$$

2. Asymptotic covariance:

$$\Sigma = \lim_{n\to\infty} n\mathsf{E}\Big[\tilde{\theta}(n)\tilde{\theta}(n)^{\intercal}\Big], \qquad \sqrt{n}\tilde{\theta}(n) \approx N(0, \Sigma)$$

Performance Criteria

Two standard approaches to evaluate performance, $\tilde{\theta}(n) := \theta(n) - \theta^*$:

1. Finite-n bound:

$$\mathsf{P}\{\|\tilde{\theta}(n)\| \geq \varepsilon\} \leq \exp(-I(\varepsilon, n)) \,, \qquad I(\varepsilon, n) = O(n\varepsilon^2)$$

2. Asymptotic covariance:

$$\Sigma = \lim_{n\to\infty} n\mathsf{E}\Big[\tilde{\theta}(n)\tilde{\theta}(n)^{\intercal}\Big] \,, \qquad \sqrt{n}\tilde{\theta}(n) \approx N(0, \Sigma)$$

Latter metric is most valuable for algorithm design.

Performance Criteria

Two standard approaches to evaluate performance, $\tilde{\theta}(n) := \theta(n) - \theta^*$:

1. Finite-n bound:

$$\mathsf{P}\{\|\tilde{\theta}(n)\| \geq \varepsilon\} \leq \exp(-I(\varepsilon, n)), \qquad I(\varepsilon, n) = O(n\varepsilon^2)$$

2. Asymptotic covariance:

$$\Sigma = \lim_{n\to\infty} n\mathsf{E}\Big[\tilde{\theta}(n)\tilde{\theta}(n)^{\intercal}\Big], \qquad \sqrt{n}\tilde{\theta}(n) \approx N(0, \Sigma)$$

Latter metric is most valuable for algorithm design.
Recall last time: $G = G^* := -A^{-1}$ then

- Resembles Monte-Carlo estimate
- Resembles Newton-Rapshon
- It is optimal: $\quad \Sigma^* = G^*\Sigma_\Delta G^{*\intercal} \leq \Sigma^G \qquad$ *any other* G

Performance Criteria

Two standard approaches to evaluate performance, $\tilde{\theta}(n) := \theta(n) - \theta^*$:

1. Finite-n bound:

$$\mathsf{P}\{\|\tilde{\theta}(n)\| \geq \varepsilon\} \leq \exp(-I(\varepsilon, n)), \qquad I(\varepsilon, n) = O(n\varepsilon^2)$$

2. Asymptotic covariance:

$$\Sigma = \lim_{n\to\infty} n\mathsf{E}\Big[\tilde{\theta}(n)\tilde{\theta}(n)^{\intercal}\Big], \qquad \sqrt{n}\tilde{\theta}(n) \approx N(0, \Sigma)$$

Latter metric is most valuable for algorithm design.
Recall last time: $G = G^* := -A^{-1}$ then

- Resembles Monte-Carlo estimate
- Resembles Newton-Rapshon *Do you see the resemblance?*
- It is optimal: $\Sigma^* = G^*\Sigma_\Delta G^{*\intercal} \leq \Sigma^G$ *any other* G

Optimal Asymptotic Covariance and Zap SNR

Resembles Newton-Rapshon?

This doesn't look much like Newton-Raphson:

$$\frac{d}{dt}x_t = -A^{-1}\bar{f}(x_t)\,, \qquad A = \frac{d}{d\theta}\bar{f}\,(\theta^*)$$

Optimal Asymptotic Covariance and Zap SNR

Zap SNR (designed to emulate deterministic Newton-Raphson)

$$\text{Requires} \quad \widehat{A}_n \approx A(\theta_n) := \frac{d}{d\theta} \bar{f}\,(\theta_n)$$

Optimal Asymptotic Covariance and Zap SNR

Zap SNR (designed to emulate Newton-Raphson)

$$\theta(n+1) = \theta(n) + \alpha_n[-\widehat{A}_n]^{-1} f(\theta(n), X(n))$$

$$\widehat{A}_n = \widehat{A}_{n-1} + \gamma_n(A_n - \widehat{A}_{n-1}), \qquad A_n = \frac{d}{d\theta} f(\theta(n), X(n))$$

Optimal Asymptotic Covariance and Zap SNR

Zap SNR (designed to emulate Newton-Raphson)

$$\theta(n+1) = \theta(n) + \alpha_n[-\widehat{A}_n]^{-1} f(\theta(n), X(n))$$

$$\widehat{A}_n = \widehat{A}_{n-1} + \gamma_n(A_n - \widehat{A}_{n-1}), \qquad A_n = \frac{d}{d\theta} f(\theta(n), X(n))$$

$$\widehat{A}_n \approx A(\theta_n) \ \text{ requires high-gain, } \frac{\gamma_n}{\alpha_n} \to \infty, \qquad n \to \infty$$

Optimal Asymptotic Covariance and Zap SNR

Zap SNR (designed to emulate Newton-Raphson)

$$\theta(n+1) = \theta(n) + \alpha_n [-\widehat{A}_n]^{-1} f(\theta(n), X(n))$$

$$\widehat{A}_n = \widehat{A}_{n-1} + \gamma_n (A_n - \widehat{A}_{n-1}), \qquad A_n = \frac{d}{d\theta} f(\theta(n), X(n))$$

$$\widehat{A}_n \approx A(\theta_n) \ \text{ requires high-gain, } \frac{\gamma_n}{\alpha_n} \to \infty, \qquad n \to \infty$$

Always: $\alpha_n = 1/n$. Numerics that follow: $\gamma_n = (1/n)^\rho$, $\rho \in (0.5, 1)$

Optimal Asymptotic Covariance and Zap SNR

Zap SNR (designed to emulate Newton-Raphson)

$$\theta(n+1) = \theta(n) + \alpha_n[-\widehat{A}_n]^{-1} f(\theta(n), X(n))$$

$$\widehat{A}_n = \widehat{A}_{n-1} + \gamma_n(A_n - \widehat{A}_{n-1}), \qquad A_n = \frac{d}{d\theta} f(\theta(n), X(n))$$

$$\widehat{A}_n \approx A(\theta_n) \text{ requires high-gain, } \frac{\gamma_n}{\alpha_n} \to \infty, \qquad n \to \infty$$

Always: $\alpha_n = 1/n$. Numerics that follow: $\gamma_n = (1/n)^\rho$, $\rho \in (0.5, 1)$

ODE for Zap SNR

$$\frac{d}{dt} x_t = -[A(x_t)]^{-1} \bar{f}(x_t), \qquad A(x) = \frac{d}{dx} \bar{f}(x)$$

Optimal Asymptotic Covariance and Zap SNR

Zap SNR (designed to emulate Newton-Raphson)

$$\theta(n+1) = \theta(n) + \alpha_n[-\widehat{A}_n]^{-1} f(\theta(n), X(n))$$

$$\widehat{A}_n = \widehat{A}_{n-1} + \gamma_n(A_n - \widehat{A}_{n-1}), \qquad A_n = \frac{d}{d\theta} f(\theta(n), X(n))$$

$$\widehat{A}_n \approx A(\theta_n) \text{ requires high-gain, } \frac{\gamma_n}{\alpha_n} \to \infty, \qquad n \to \infty$$

Always: $\alpha_n = 1/n$. Numerics that follow: $\gamma_n = (1/n)^\rho$, $\rho \in (0.5, 1)$

ODE for Zap SNR

$$\frac{d}{dt} x_t = -[A(x_t)]^{-1} \bar{f}(x_t), \qquad A(x) = \frac{d}{dx} \bar{f}(x)$$

- Not necessarily stable
- *General* conditions for convergence is open

Reinforcement Learning and Stochastic Approximation

SA and RL Design

Functional equations in Stochastic Control

Always of the form

$$0 = \mathsf{E}[F(h^*, \Phi(n+1)) \mid \Phi(0) \dots \Phi(n)]\,, \qquad h^* = ?$$

SA and RL Design

Functional equations in Stochastic Control

Always of the form

$$0 = \mathsf{E}[F(h^*, \Phi(n+1)) \mid \Phi(0) \ldots \Phi(n)]\,, \qquad h^* = ?$$

$$\Phi(n) = (\text{state, action})$$

SA and RL Design

Functional equations in Stochastic Control

Always of the form

$$0 = \mathsf{E}[F(h^*, \Phi(n+1)) \mid \Phi(0) \dots \Phi(n)]\,, \qquad h^* = ?$$

Galerkin relaxation:

$$0 = \mathsf{E}[F(h^{\theta^*}, \Phi(n+1))\zeta_n]\,, \qquad \theta^* = ?$$

SA and RL Design

Functional equations in Stochastic Control

Always of the form

$$0 = \mathsf{E}[F(h^*, \Phi(n+1)) \mid \Phi(0) \dots \Phi(n)]\,, \qquad h^* = ?$$

Galerkin relaxation:

$$0 = \mathsf{E}[F(h^{\theta^*}, \Phi(n+1))\zeta_n]\,, \qquad \theta^* = ?$$

Necessary Ingredients:

- Parameterized family $\{h^\theta : \theta \in \mathbb{R}^d\}$
- Adapted, d-dimensional stochastic process $\{\zeta_n\}$

Examples are TD- and Q-Learning

SA and RL Design

Functional equations in Stochastic Control

Always of the form

$$0 = \mathsf{E}[F(h^*, \Phi(n+1)) \mid \Phi(0) \dots \Phi(n)] \,, \qquad h^* = ?$$

Galerkin relaxation:

$$0 = \mathsf{E}[F(h^{\theta^*}, \Phi(n+1))\zeta_n] \,, \qquad \theta^* = ?$$

Necessary Ingredients:

- Parameterized family $\{h^\theta : \theta \in \mathbb{R}^d\}$
- Adapted, d-dimensional stochastic process $\{\zeta_n\}$

Examples are TD- and Q-Learning

These algorithms are thus special cases of stochastic approximation

Stochastic Optimal Control

MDP Model

$\boldsymbol{X}$ is a controlled Markov chain, with input $\boldsymbol{U}$

- For all states x and sets A,

  $\mathsf{P}\{X(n+1) \in A \mid X(n) = x,\; U(n) = u, \text{and prior history}\} = P_u(x, A)$
- $c\colon \mathsf{X} \times \mathsf{U} \to \mathbb{R}$ is a cost function
- $\beta < 1$ a discount factor

restrict to finite state and action space here

Stochastic Optimal Control

MDP Model

$\boldsymbol{X}$ is a controlled Markov chain, with input $\boldsymbol{U}$

- For all states x and sets A,

  $\mathsf{P}\{X(n+1) \in A \mid X(n) = x,\ U(n) = u, \text{and prior history}\} = P_u(x, A)$
- $c \colon \mathsf{X} \times \mathsf{U} \to \mathbb{R}$ is a cost function
- $\beta < 1$ a discount factor

restrict to finite state and action space here

Value function:

$$h^*(x) = \min_{\boldsymbol{U}} \sum_{n=0}^{\infty} \beta^n \mathsf{E}[c(X(n), U(n)) \mid X(0) = x]$$

Stochastic Optimal Control

MDP Model

$\boldsymbol{X}$ is a controlled Markov chain, with input $\boldsymbol{U}$

- For all states x and sets A,

  $\mathsf{P}\{X(n+1) \in A \mid X(n) = x,\ U(n) = u, \text{and prior history}\} = P_u(x, A)$
- $c\colon \mathsf{X} \times \mathsf{U} \to \mathbb{R}$ is a cost function
- $\beta < 1$ a discount factor

restrict to finite state and action space here

Value function:

$$h^*(x) = \min_{\boldsymbol{U}} \sum_{n=0}^{\infty} \beta^n \mathsf{E}[c(X(n), U(n)) \mid X(0) = x]$$

Bellman equation:

$$h^*(x) = \min_u \{c(x,u) + \beta \mathsf{E}[h^*(X(n+1)) \mid X(n) = x,\ U(n) = u]\}$$

Q-function

Trick to swap expectation and minimum

Bellman equation:

$$h^*(x) = \min_u \{c(x,u) + \beta \mathsf{E}[h^*(X(n+1)) \mid X(n) = x,\ U(n) = u]\}$$

Q-function

Trick to swap expectation and minimum

Bellman equation:

$$h^*(x) = \min_u \{ c(x,u) + \beta \mathsf{E}[h^*(X(n+1)) \mid X(n) = x,\ U(n) = u] \}$$

Q-function:

$$Q^*(x,u) := c(x,u) + \beta \mathsf{E}[h^*(X(n+1)) \mid X(n) = x,\ U(n) = u]$$

Q-function

Trick to swap expectation and minimum

Bellman equation:

$$h^*(x) = \min_u \{c(x,u) + \beta \mathsf{E}[h^*(X(n+1)) \mid X(n) = x,\ U(n) = u]\}$$

Q-function:

$$Q^*(x,u) := c(x,u) + \beta \mathsf{E}[h^*(X(n+1)) \mid X(n) = x,\ U(n) = u]$$

$$h^*(x) = \min_u Q^*(x,u)$$

Q-function

Trick to swap expectation and minimum

Bellman equation:

$$h^*(x) = \min_u \{c(x,u) + \beta \mathsf{E}[h^*(X(n+1)) \mid X(n) = x,\ U(n) = u]\}$$

Q-function:

$$Q^*(x,u) := c(x,u) + \beta \mathsf{E}[h^*(X(n+1)) \mid X(n) = x,\ U(n) = u]$$

$$h^*(x) = \min_u Q^*(x,u)$$

Another Bellman equation:

$$Q^*(x,u) = c(x,u) + \beta \mathsf{E}[\underline{Q}^*(X(n+1)) \mid X(n) = x,\ U(n) = u]$$

$$\underline{Q}^*(x) = \min_u Q^*(x,u)$$

Q-function

Trick to swap expectation and minimum

Another Bellman equation:

$$Q^*(x,u) = c(x,u) + \beta \mathsf{E}[\underline{Q}^*(X(n+1)) \mid X(n) = x,\ U(n) = u]$$

$$\underline{Q}^*(x) = \min_u Q^*(x,u)$$

$$Q^*(x,u) = \min_{\boldsymbol{U}} \sum_{n=0}^{\infty} \beta^n \mathsf{E}[c(X(n),U(n)) \mid X(0) = x, U(0) = u]$$

Q-function

Trick to swap expectation and minimum

Another Bellman equation:

$$Q^*(x,u) = c(x,u) + \beta \mathsf{E}[\underline{Q}^*(X(n+1)) \mid X(n) = x,\ U(n) = u]$$

$$\underline{Q}^*(x) = \min_u Q^*(x,u)$$

$$Q^*(x,u) = \min_{\boldsymbol{U}} \sum_{n=0}^{\infty} \beta^n \mathsf{E}[c(X(n),U(n)) \mid X(0) = x, U(0) = u]$$

One-to-one mapping between cost functions and Q-functions.
Notation:

$$Q^* = \mathcal{Q}^*(c)$$

Q-Learning and Galerkin Relaxation

Dynamic programming

Find function Q^* that solves

$$\mathsf{E}\Big[c(X(n),U(n)) + \beta \underline{Q}^*(X(n+1)) - Q^*(X(n),U(n)) \mid \mathcal{F}_n\Big] = 0$$

Q-Learning and Galerkin Relaxation

Dynamic programming

Find function Q^* that solves

$$\mathsf{E}\Big[c(X(n),U(n)) + \beta \underline{Q}^*(X(n+1)) - Q^*(X(n),U(n)) \mid \mathcal{F}_n\Big] = 0$$

That is,

$$0 = \mathsf{E}[F(Q^*, \Phi(n+1)) \mid \Phi(0) \; \dots \; \Phi(n)]\,,$$

$$\text{with } \Phi(n+1) = (X(n+1), X(n), U(n)).$$

Q-Learning and Galerkin Relaxation

Dynamic programming

Find function Q^* that solves

$$\mathsf{E}\Big[c(X(n),U(n)) + \beta \underline{Q}^*(X(n+1)) - Q^*(X(n),U(n)) \mid \mathcal{F}_n\Big] = 0$$

Q-Learning

Find θ^* that solves

$$\mathsf{E}\Big[\big(c(X(n),U(n)) + \beta \underline{Q}^{\theta^*}((X(n+1)) - Q^{\theta^*}((X(n),U(n))\big)\zeta_n\Big] = 0$$

where the input $\boldsymbol{U}$ is randomized state feedback

Q-Learning and Galerkin Relaxation

Dynamic programming

Find function Q^* that solves

$$\mathsf{E}\big[c(X(n),U(n)) + \beta \underline{Q}^*(X(n+1)) - Q^*(X(n),U(n)) \mid \mathcal{F}_n\big] = 0$$

Q-Learning

Find θ^* that solves

$$\mathsf{E}\Big[\big(c(X(n),U(n)) + \beta \underline{Q}^{\theta^*}((X(n+1)) - Q^{\theta^*}((X(n),U(n))\big)\zeta_n\Big] = 0$$

where the input $\boldsymbol{U}$ is randomized state feedback

The family $\{Q^\theta\}$ and *eligibility vectors* $\{\zeta_n\}$ are part of algorithm design.

Watkins' Q-learning

Find θ^* that solves

$$\mathsf{E}\Big[\Big(c(X(n),U(n)) + \beta \underline{Q}^{\theta^*}((X(n+1)) - Q^{\theta^*}((X(n),U(n))\Big)\zeta_n\Big] = 0$$

Watkins' Q-learning

Find θ^* that solves

$$\mathsf{E}\Big[\Big(c(X(n),U(n)) + \beta \underline{Q}^{\theta^*}((X(n+1)) - Q^{\theta^*}((X(n),U(n))\Big)\zeta_n\Big] = 0$$

Watkin's algorithm is Stochastic Approximation

The family $\{Q^\theta\}$ and *eligibility vectors* $\{\zeta_n\}$ in this design:

- Linearly parameterized family of functions: $Q^\theta(x,u) = \theta^\intercal \psi(x,u)$
- $\zeta_n \equiv \psi(X_n, U_n)$ *and*
- $\psi_n(x,u) = 1\{x = x^n, u = u^n\}$ (complete basis)

Watkins' Q-learning

Find θ^* that solves

$$\mathsf{E}\Big[\Big(c(X(n),U(n)) + \beta \underline{Q}^{\theta^*}((X(n+1)) - Q^{\theta^*}((X(n),U(n))\Big)\zeta_n\Big] = 0$$

Watkin's algorithm is Stochastic Approximation

The family $\{Q^\theta\}$ and *eligibility vectors* $\{\zeta_n\}$ in this design:

- Linearly parameterized family of functions: $Q^\theta(x,u) = \theta^{\intercal}\psi(x,u)$
- $\zeta_n \equiv \psi(X_n, U_n)$ *and*
- $\psi_n(x,u) = 1\{x = x^n, u = u^n\}$ (complete basis)

Asymptotic covariance is typically infinite

Watkins' Q-learning

Big Question: *Can we Zap Q-Learning?*

Find θ^* that solves

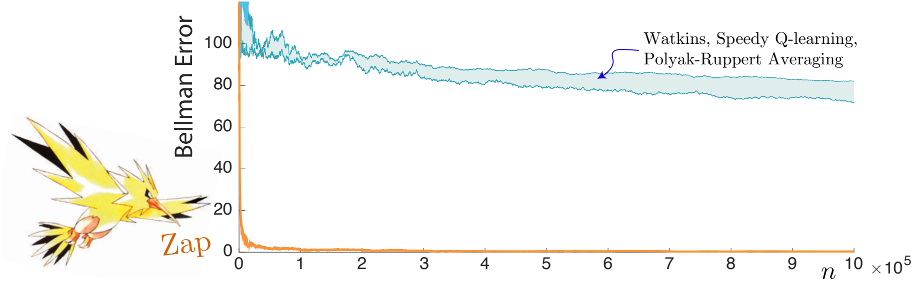
$$\mathsf{E}\Big[\big(c(X(n),U(n)) + \beta \underline{Q}^{\theta^*}((X(n+1)) - Q^{\theta^*}((X(n),U(n))\big)\zeta_n\Big] = 0$$

Watkin's algorithm is Stochastic Approximation

The family $\{Q^\theta\}$ and *eligibility vectors* $\{\zeta_n\}$ in this design:

- Linearly parameterized family of functions: $Q^\theta(x,u) = \theta^{\intercal}\psi(x,u)$
- $\zeta_n \equiv \psi(X_n, U_n)$ *and*
- $\psi_n(x,u) = 1\{x = x^n, u = u^n\}$ (complete basis)

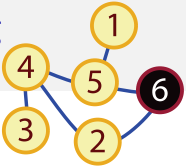
Asymptotic covariance is typically infinite



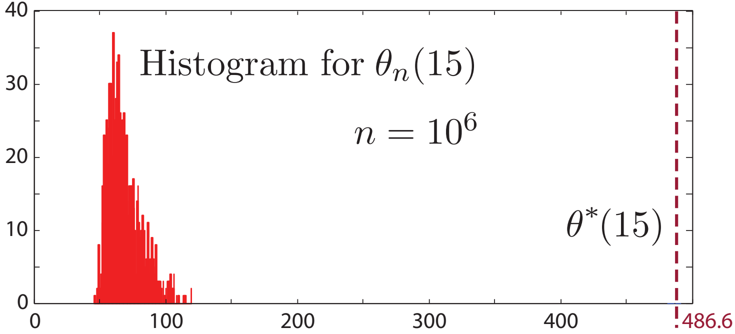
Zap Q-Learning

Asymptotic Covariance of Watkins' Q-Learning

Improvements are needed!



Histogram of parameter estimates after 10^6 iterations.



Example from Devraj & M 2017

Zap Q-learning

Zap Q-Learning $\equiv$ Zap SNR for Q-Learning

$$
\begin{aligned}
0 = \bar{f}(\theta) &= \mathsf{E}\big[f(\theta, \Phi(n+1))\big] \\
&:= \mathsf{E}\Big[\zeta_n \big[c(X(n), U(n)) + \beta \underline{Q}^\theta(X(n+1)) - Q^\theta(X(n), U(n))\big]\Big]
\end{aligned}
$$

Zap Q-learning

Zap Q-Learning $\equiv$ Zap SNR for Q-Learning

$$0 = \bar{f}(\theta) = \mathsf{E}\big[f(\theta, \Phi(n+1))\big]$$
$$:= \mathsf{E}\Big[\zeta_n\Big[c(X(n),U(n)) + \beta \underline{Q}^{\theta}(X(n+1)) - Q^{\theta}(X(n),U(n))\Big]\Big]$$

- $A(\theta) = \frac{d}{d\theta}\bar{f}(\theta)$;

Zap Q-learning

Zap Q-Learning ≡ Zap SNR for Q-Learning

$$0 = \bar{f}(\theta) = \mathsf{E}\big[f(\theta, \Phi(n+1))\big]$$
$$:= \mathsf{E}\Big[\zeta_n\big[c(X(n), U(n)) + \beta \underline{Q}^\theta(X(n+1)) - Q^\theta(X(n), U(n))\big]\Big]$$

- $A(\theta) = \frac{d}{d\theta}\bar{f}(\theta)$; At points of differentiability:

$$A(\theta) = \mathsf{E}\Big[\zeta_n\big[\beta\psi(X(n+1), \phi^\theta(X(n+1))) - \psi(X(n), U(n))\big]^{\mathsf{T}}\Big]$$
$$\phi^\theta(X(n+1)) := \arg\min_u Q^\theta(X(n+1), u)$$

Zap Q-learning

Zap Q-Learning $\equiv$ Zap SNR for Q-Learning

$$0 = \bar{f}(\theta) = \mathsf{E}\big[f(\theta, \Phi(n+1))\big]$$
$$:= \mathsf{E}\Big[\zeta_n \big[c(X(n), U(n)) + \beta \underline{Q}^{\theta}(X(n+1)) - Q^{\theta}(X(n), U(n))\big]\Big]$$

- $A(\theta) = \frac{d}{d\theta}\bar{f}(\theta)$; At points of differentiability:

$$A(\theta) = \mathsf{E}\Big[\zeta_n \big[\beta\psi(X(n+1), \phi^{\theta}(X(n+1))) - \psi(X(n), U(n))\big]^{\mathsf{T}}\Big]$$
$$\phi^{\theta}(X(n+1)) := \arg\min_u Q^{\theta}(X(n+1), u)$$

Algorithm:

$$\theta(n+1) = \theta(n) + \alpha_n\big[-\widehat{A}_n\big]^{-1} f(\theta(n), \Phi(n+1)); \qquad \widehat{A}_n = \widehat{A}_{n-1} + \gamma_n(A_n - \widehat{A}_{n-1});$$

Zap Q-learning

Zap Q-Learning $\equiv$ Zap SNR for Q-Learning

$$0 = \bar{f}(\theta) = \mathsf{E}\big[f(\theta, \Phi(n+1))\big]$$
$$:= \mathsf{E}\Big[\zeta_n \big[c(X(n), U(n)) + \beta \underline{Q}^{\theta}(X(n+1)) - Q^{\theta}(X(n), U(n))\big]\Big]$$

- $A(\theta) = \frac{d}{d\theta}\bar{f}(\theta)$; At points of differentiability:

$$A(\theta) = \mathsf{E}\Big[\zeta_n \big[\beta\psi(X(n+1), \phi^{\theta}(X(n+1))) - \psi(X(n), U(n))\big]^{\mathsf{T}}\Big]$$
$$\phi^{\theta}(X(n+1)) := \arg\min_u Q^{\theta}(X(n+1), u)$$

Algorithm:

$$\theta(n+1) = \theta(n) + \alpha_n\big[-\widehat{A}_n\big]^{-1} f(\theta(n), \Phi(n+1)); \qquad \widehat{A}_n = \widehat{A}_{n-1} + \gamma_n(A_n - \widehat{A}_{n-1});$$
$$A_{n+1} := \frac{d}{d\theta} f(\theta_n, \Phi(n+1))$$
$$= \zeta_n \big[\beta\psi(X(n+1), \phi^{\theta_n}(X(n+1))) - \psi(X(n), U(n))\big]^{\mathsf{T}}$$

Zap Q-learning

Zap Q-Learning $\equiv$ Zap SNR for Q-Learning

$$0 = \bar{f}(\theta) = \mathsf{E}\big[f(\theta, \Phi(n+1))\big]$$
$$:= \mathsf{E}\Big[\zeta_n\big[c(X(n), U(n)) + \beta \underline{Q}^\theta(X(n+1)) - Q^\theta(X(n), U(n))\big]\Big]$$

- $A(\theta) = \frac{d}{d\theta}\bar{f}(\theta)$; At points of differentiability:

$$A(\theta) = \mathsf{E}\Big[\zeta_n\big[\beta\psi(X(n+1), \phi^\theta(X(n+1))) - \psi(X(n), U(n))\big]^{\mathsf{T}}\Big]$$
$$\phi^\theta(X(n+1)) := \arg\min_u Q^\theta(X(n+1), u)$$

Algorithm:

$$\theta(n+1) = \theta(n) + \alpha_n\big[-\widehat{A}_n\big]^{-1} f(\theta(n), \Phi(n+1)); \qquad \widehat{A}_n = \widehat{A}_{n-1} + \gamma_n(A_n - \widehat{A}_{n-1});$$
$$A_{n+1} := \frac{d}{d\theta} f(\theta_n, \Phi(n+1))$$
$$= \zeta_n\big[\beta\psi(X(n+1), \phi^{\theta_n}(X(n+1))) - \psi(X(n), U(n))\big]^{\mathsf{T}}$$

Stable?

Zap Q-learning

Zap Q-Learning ≡ Zap SNR for Q-Learning

ODE Analysis: change of variables $q = \mathcal{Q}^*(\varsigma)$

Functional $\mathcal{Q}^*$ maps cost functions to Q-functions:

$$q(x,u) = \varsigma(x,u) + \beta \sum_{x'} P_u(x,x') \min_{u'} q(x',u')$$

Zap Q-learning

Zap Q-Learning $\equiv$ Zap SNR for Q-Learning

ODE Analysis: change of variables $q = \mathcal{Q}^*(\varsigma)$

Functional $\mathcal{Q}^*$ maps cost functions to Q-functions:

$$q(x,u) = \varsigma(x,u) + \beta \sum_{x'} P_u(x,x') \min_{u'} q(x',u')$$

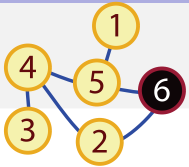
ODE for Zap-Q

$$q_t = \mathcal{Q}^*(\varsigma_t), \qquad \frac{d}{dt}\varsigma_t = -\varsigma_t + c$$

$\Rightarrow$ convergence, optimal covariance, ...

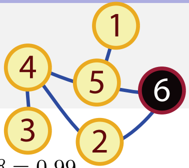
Zap Q-Learning

Example: Optimize Walk to Cafe



Zap Q-Learning

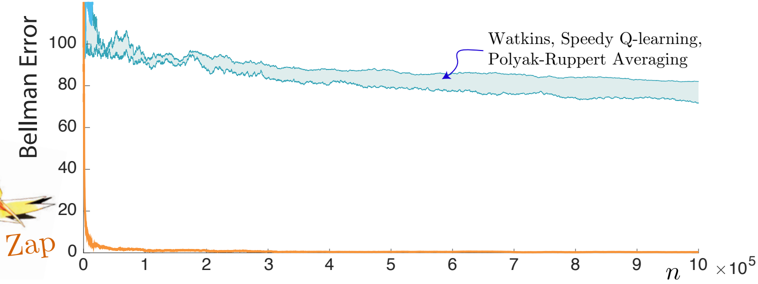
Example: Optimize Walk to Cafe



Convergence with Zap gain $\gamma_n = n^{-0.85}$

Discount factor: $\beta = 0.99$

Watkins' algorithm has infinite asymptotic covariance with $\alpha_n = 1/n$

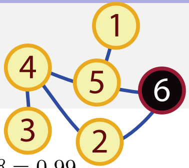


Convergence of Zap-Q Learning

Discount factor: $\beta = 0.99$

Zap Q-Learning

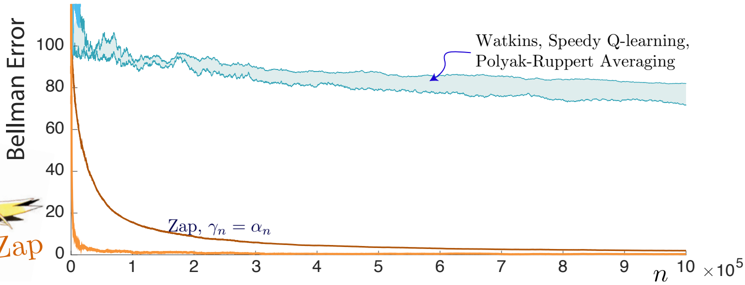
Example: Optimize Walk to Cafe



Convergence with Zap gain $\gamma_n = n^{-0.85}$

Discount factor: $\beta = 0.99$

Watkins' algorithm has infinite asymptotic covariance with $\alpha_n = 1/n$

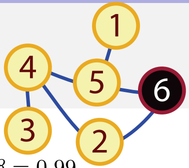


Convergence of Zap-Q Learning

Discount factor: $\beta = 0.99$

Zap Q-Learning

Example: Optimize Walk to Cafe

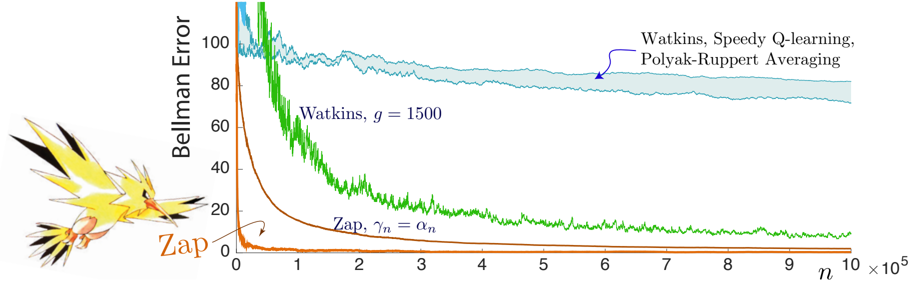


Convergence with Zap gain $\gamma_n = n^{-0.85}$

Discount factor: $\beta = 0.99$

Watkins' algorithm has infinite asymptotic covariance with $\alpha_n = 1/n$

Optimal scalar gain is approximately $\alpha_n = 1500/n$

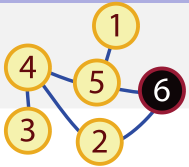


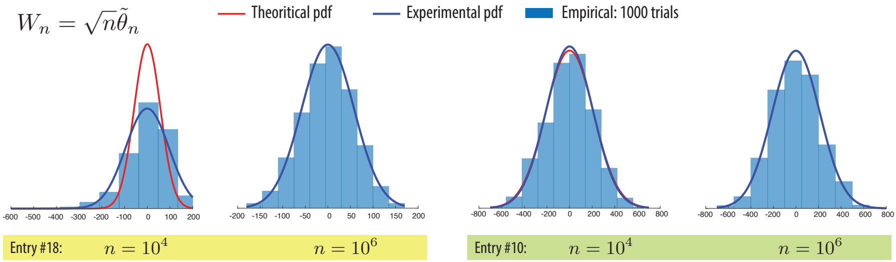
Convergence of Zap-Q Learning

Discount factor: $\beta = 0.99$

Zap Q-Learning

Example: Optimize Walk to Cafe

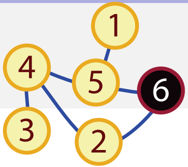




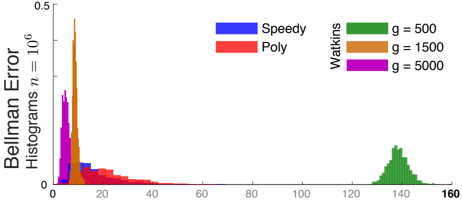
CLT gives good prediction of finite-n performance

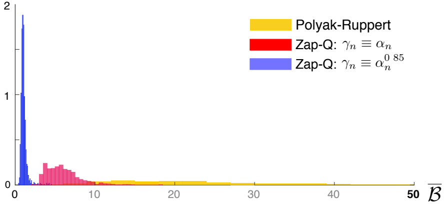
Zap Q-Learning

Example: Optimize Walk to Cafe



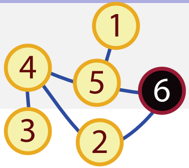
Local Convergence: $\theta(0)$ initialized in neighborhood of θ^*



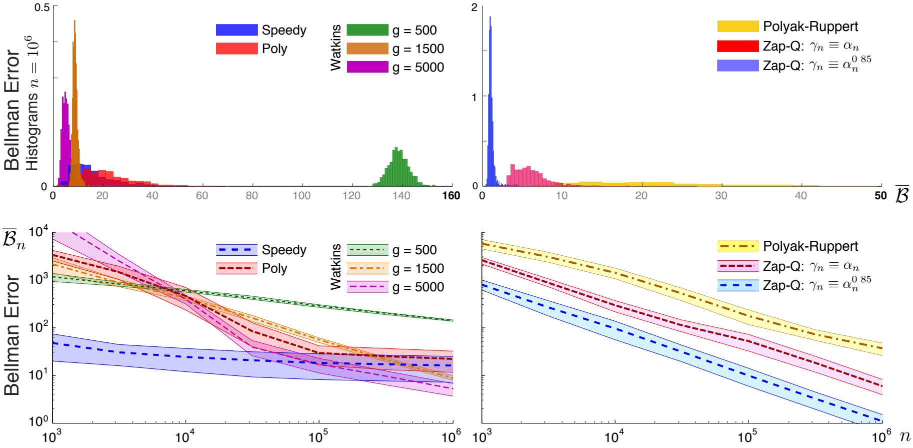


Zap Q-Learning

Example: Optimize Walk to Cafe



Local Convergence: $\theta(0)$ initialized in neighborhood of θ^*



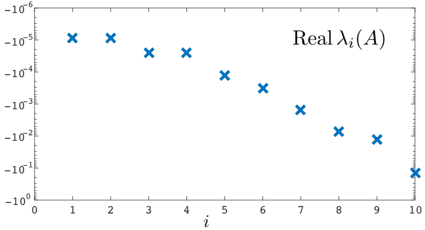
2σ confidence intervals for the Q-learning algorithms

Zap Q-Learning

Model of Tsitsiklis and Van Roy: **Optimal Stopping Time in Finance**

State space: $\mathbb{R}^{100}$

Parameterized Q-function: Q^θ with $\theta \in \mathbb{R}^{10}$



Real $\lambda > -\frac{1}{2}$ for every eigenvalue λ

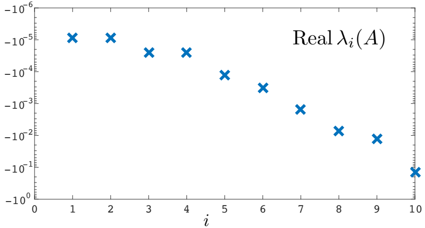
Asymptotic covariance is infinite

Zap Q-Learning

Model of Tsitsiklis and Van Roy: **Optimal Stopping Time in Finance**

State space: $\mathbb{R}^{100}$

Parameterized Q-function: Q^θ with $\theta \in \mathbb{R}^{10}$



Real $\lambda > -\frac{1}{2}$ for every eigenvalue λ

Asymptotic covariance is infinite

Authors observed slow convergence
Proposed a matrix gain sequence

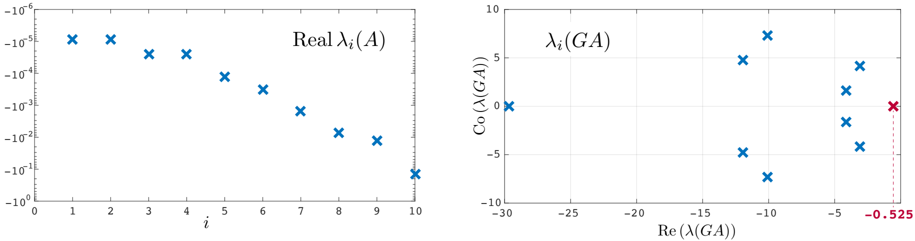
$\{G_n\}$ (see refs for details)

Zap Q-Learning

Model of Tsitsiklis and Van Roy: **Optimal Stopping Time in Finance**

State space: $\mathbb{R}^{100}$

Parameterized Q-function: Q^θ with $\theta \in \mathbb{R}^{10}$



Eigenvalues of A and GA for the finance example

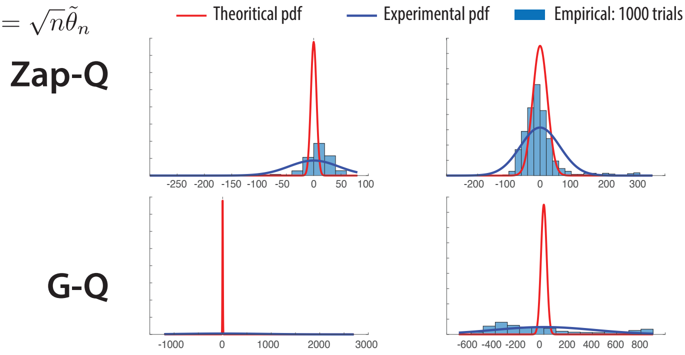
Favorite choice of gain in [25] barely meets the criterion Re$(\lambda(GA)) < -\frac{1}{2}$

Zap Q-Learning

Model of Tsitsiklis and Van Roy: **Optimal Stopping Time in Finance**

State space: $\mathbb{R}^{100}$. Parameterized Q-function: Q^θ with $\theta \in \mathbb{R}^{10}$

$$W_n = \sqrt{n}\tilde{\theta}_n$$



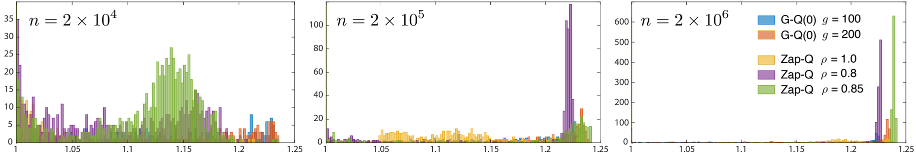
Entry #1: $n = 2 \times 10^6$ Entry #7: $n = 2 \times 10^6$

Zap Q-Learning

Model of Tsitsiklis and Van Roy: **Optimal Stopping Time in Finance**

State space: $\mathbb{R}^{100}$.
Parameterized Q-function: Q^θ with $\theta \in \mathbb{R}^{10}$

Histograms of the average reward obtained using the different algorithms:



Zap-Q $\gg$ G-Q

Conclusions

Conclusions & Future Work

Conclusions

- The asymptotic covariance is an awesome design tool.
  It is also predictive of finite-n performance.

  Example: $g^* = 1500$ was chosen based on **asymptotic** covariance

Conclusions & Future Work

Conclusions

- The asymptotic covariance is an awesome design tool.
  It is also predictive of finite-n performance.

  Example: $g^* = 1500$ was chosen based on **asymptotic** covariance

- The success of Zap Q-Learning is due to two factors:
  - Choice of gain for optimal asymptotic variance (validated in simulations)
  - Luck: Newton-Raphson is globally stable

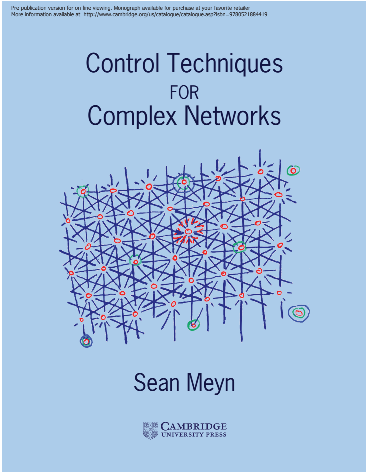
Conclusions & Future Work

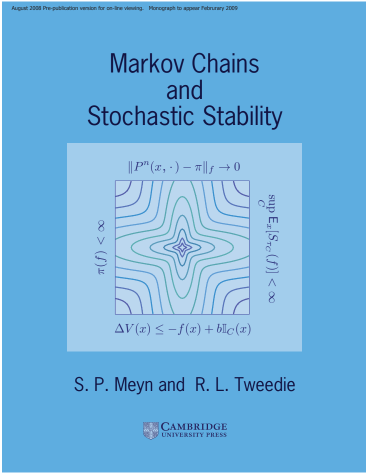
Conclusions

- The success of Zap Q-Learning is due to two factors:
  - Choice of gain for optimal asymptotic variance (validated in simulations)
  - Luck: Newton-Raphson is globally stable

- Future work:
  - Q-learning with function-approximation
    - *Obtain conditions for a stable algorithm in a general setting*
    - *Optimal stopping time problems*
  - Reduced complexity algorithms with adaptive optimization of algorithm parameters (*stay tuned for revision on arXiv*)

Thank you!


Pre-publication version for on-line viewing. Monograph available for purchase at your favorite retailer
More information available at http://www.cambridge.org/us/catalogue/catalogue.asp?isbn=9780521884419
Control Techniques
FOR
Complex Networks
Sean Meyn
CAMBRIDGE
UNIVERSITY PRESS


August 2008 Pre-publication version for on-line viewing. Monograph to appear Februrary 2009
Markov Chains
and
Stochastic Stability
$\|P^n(x,\cdot)-\pi\|_f \to 0$
$\pi(f) < \infty$
$\sup_C \mathsf{E}_x[S_{\tau_C}(f)] < \infty$
$\Delta V(x) \leq -f(x) + b\mathbb{I}_C(x)$
S. P. Meyn and R. L. Tweedie
CAMBRIDGE
UNIVERSITY PRESS

References

Selected References I

[1] V. S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Hindustan Book Agency and Cambridge University Press (jointly), Delhi, India and Cambridge, UK, 2008.

[2] A. M. Devraj and S. P. Meyn. *Fastest convergence for Q-learning.* *ArXiv* , July 2017.

[3] M. Benaïm. *Dynamics of stochastic approximation algorithms.* In *Séminaire de Probabilités XXXIII*, pages 1–68, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[4] V. S. Borkar and S. P. Meyn. *The ODE method for convergence of stochastic approximation and reinforcement learning.* *SIAM J. Control Optim.*, 38(2):447–469, 2000.

[5] A. Benveniste, M. Métivier, and P. Priouret. *Adaptive algorithms and stochastic approximations*, volume 22 of *Applications of Mathematics*. Springer-Verlag, Berlin, 1990.

[6] P. J. Schweitzer. *Perturbation theory and finite Markov chains.* *J. Appl. Prob.*, 5:401–403, 1968.

[7] S. P. Meyn and R. L. Tweedie. *Markov chains and stochastic stability*. Cambridge University Press, Cambridge, second edition, 2009. Cambridge Mathematical Library.

[8] S. P. Meyn. *Control Techniques for Complex Networks*. Cambridge University Press, 2007. *See last chapter on simulation and average-cost TD learning*

Selected References II

[9] D. Ruppert. *A Newton-Raphson version of the multivariate Robbins-Monro procedure.* *The Annals of Statistics*, 13(1):236–245, 1985.

[10] D. Ruppert. *Efficient estimators from a slowly convergent Robbins-Monro processes.* Technical Report Tech. Rept. No. 781, Cornell University, School of Operations Research and Industrial Engineering, Ithaca, NY, 1988.

[11] B. T. Polyak. *A new method of stochastic approximation type.* *Avtomatika i telemekhanika (in Russian). translated in Automat. Remote Control, 51 (1991)*, pages 98–107, 1990.

[12] B. T. Polyak and A. B. Juditsky. *Acceleration of stochastic approximation by averaging.* *SIAM J. Control Optim.*, 30(4):838–855, 1992.

[13] V. R. Konda and J. N. Tsitsiklis. *Convergence rate of linear two-time-scale stochastic approximation.* *Ann. Appl. Probab.*, 14(2):796–819, 2004.

[14] E. Moulines and F. R. Bach. *Non-asymptotic analysis of stochastic approximation algorithms for machine learning.* In *Advances in Neural Information Processing Systems 24*, pages 451–459. Curran Associates, Inc., 2011.

Selected References III

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

[16] C. J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge, Cambridge, UK, 1989.

[17] C. J. C. H. Watkins and P. Dayan. *Q-learning*. *Machine Learning*, 8(3-4):279–292, 1992.

[18] R. S. Sutton.*Learning to predict by the methods of temporal differences*. *Mach. Learn.*, 3(1):9–44, 1988.

[19] J. N. Tsitsiklis and B. Van Roy. *An analysis of temporal-difference learning with function approximation*. *IEEE Trans. Automat. Control*, 42(5):674–690, 1997.

[20] C. Szepesvári. *The asymptotic convergence-rate of Q-learning*. In *Proceedings of the 10th Internat. Conf. on Neural Info. Proc. Systems*, pages 1064–1070. MIT Press, 1997.

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

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

Selected References IV

[23] D. Huang, W. Chen, P. Mehta, S. Meyn, and A. Surana. *Feature selection for neuro-dynamic programming.* In F. Lewis, editor, *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*. Wiley, 2011.

[24] 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*, 44(10):1840–1851, 1999.

[25] 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*, 16(2):207–239, 2006.

[26] S. J. Bradtke and A. G. Barto. *Linear least-squares algorithms for temporal difference learning. Mach. Learn.*, 22(1-3):33–57, 1996.

[27] J. A. Boyan. *Technical update: Least-squares temporal difference learning. Mach. Learn.*, 49(2-3):233–246, 2002.

[28] A. Nedic and D. Bertsekas. *Least squares policy evaluation algorithms with linear function approximation. Discrete Event Dyn. Systems: Theory and Appl.*, 13(1-2):79–110, 2003.

[29] P. G. Mehta and S. P. Meyn. *Q-learning and Pontryagin's minimum principle.* In *IEEE Conference on Decision and Control*, pages 3598–3605, Dec. 2009.