Stochastic Approximation and Reinforcement Learning

Hidden Theory and New Super-Fast Algorithms

Microsoft Research, Redmond — October 9, 2018

Adithya M. Devraj

Department of Electrical and Computer Engineering — University of Florida

Thanks to to the National Science Foundation & University of Florida Informatics Institute

Based on joint work with Ana Bušić @ Inria and Sean Meyn @ UF

Stochastic Approximation and Reinforcement Learning

Outline

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

Stochastic Approximation

What is Stochastic Approximation?

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 variable W may not be known
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 [D & Meyn 2017]

Algorithm & Convergence

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

Algorithm (Robbins & Monro 1951):

$$\theta(n+1) = \theta(n) + \alpha_{n+1} f(\theta(n), W(n+1))$$

The step-size satisfies

- $\sum \alpha_n = \infty \qquad \sum \alpha_n^2 < \infty$
- Usually we will take $\alpha_n = 1/n$

Rewriting the recursion:

$$\theta(n+1) = \theta(n) + \alpha_{n+1}[\bar{f}(\theta(n)) + \Delta_{n+1}]$$

Interpretation: $\theta^* \equiv$ *stationary point* of the ODE

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

Analysis: Stability of the ODE $\oplus$ (See Borkar's monograph) $\Longrightarrow$

$$\lim_{n \to \infty} \theta(n) = \theta^*$$

Stochastic Approximation Example

Monte-Carlo

Monte-Carlo Estimation

Estimate the mean $\eta = \mathsf{E}[c(W)]$, where random variable W has density ϱ:

$$\eta = \int c(w)\, \varrho(w)\, dx$$

Stochastic Approximation Example

Monte-Carlo

Monte-Carlo Estimation

Estimate the mean $\eta = \mathsf{E}[c(W)]$

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

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

Stochastic Approximation Example

Monte-Carlo

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

Monte-Carlo Estimation

Estimate the mean $\eta = \mathsf{E}[c(W)]$

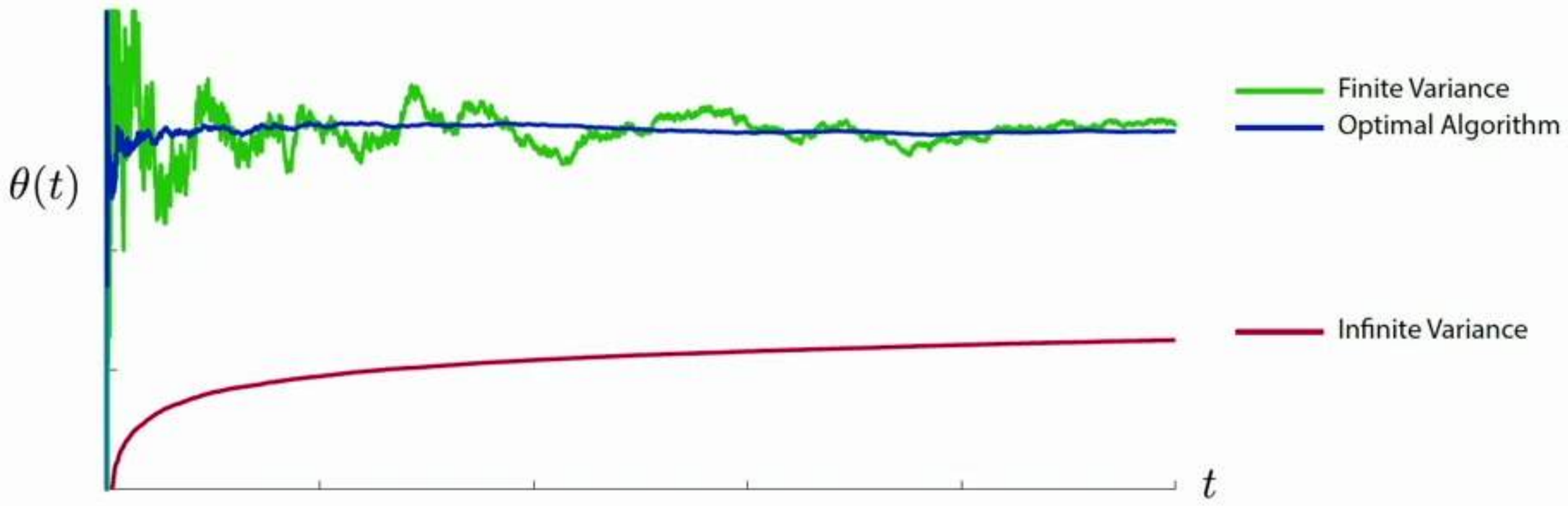
SA interpretation: Find θ^* solving $0 = \mathsf{E}[f(\theta, W)] = \mathsf{E}[c(W) - \theta]$

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

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

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

SA Recursion: $\theta(n+1) = \theta(n) + \alpha_{n+1} f(\theta(n), W(n+1))$



Fastest Stochastic Approximation

Performance Criteria

Error sequence:

$$\tilde{\theta}(n) := \theta(n) - \theta^*$$

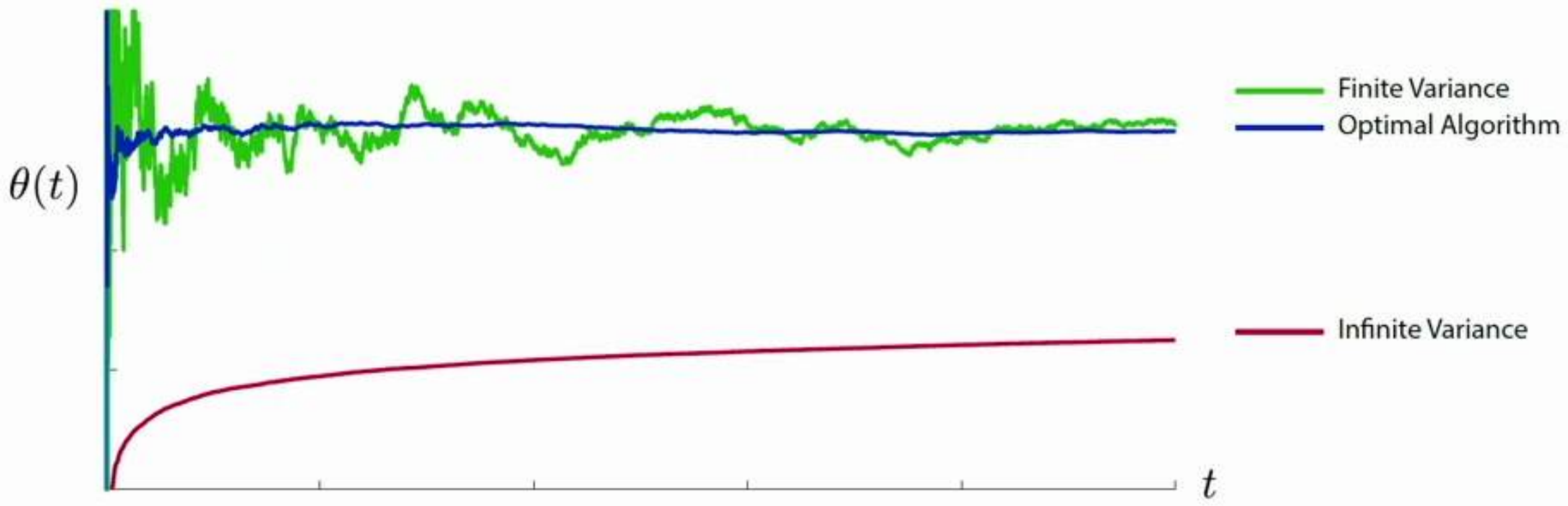
Two standard approaches to evaluate performance,

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:

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



Fastest Stochastic Approximation

Performance Criteria

Error sequence:

$$\tilde{\theta}(n) := \theta(n) - \theta^*$$

Two standard approaches to evaluate performance,

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:

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

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 $\{\Sigma_n\}$

$$\theta(n+1) = \theta(n) + \alpha_{n+1}[\bar{f}(\theta(n)) + \Delta_{n+1}]$$

Performance Criteria

Error sequence:

$$\tilde{\theta}(n) := \theta(n) - \theta^*$$

Two standard approaches to evaluate performance,

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:

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

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 $\{\Sigma_n\}$

$$\theta(n+1) = \theta(n) + \alpha_{n+1}[\bar{f}(\theta(n)) + \Delta_{n+1}]$$

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 $\{\Sigma_n\}$

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

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

$$\Sigma_\Delta = \mathsf{E}[\Delta_{n+1}\Delta_{n+1}^{\intercal}]$$

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 $\{\Sigma_n\}$

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

$$A = \tfrac{d}{d\theta}\bar{f}\,(\theta^*)$$
$$\Sigma_\Delta = \mathsf{E}[\Delta_{n+1}\Delta_{n+1}^{\intercal}]$$

Asymptotic Variance Theory

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

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

Stochastic Approximation Example

Monte-Carlo

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

Monte-Carlo Estimation

Estimate the mean $\eta = \mathsf{E}[c(W)]$

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

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

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

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

SA Recursion: $\theta(n+1) = \theta(n) + \alpha_{n+1} f(\theta(n), W(n+1))$

Algorithm & Convergence

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

Algorithm (Robbins & Monro 1951):

$$\theta(n+1) = \theta(n) + \alpha_{n+1} f(\theta(n), W(n+1))$$

The step-size satisfies

- $\sum \alpha_n = \infty \qquad \sum \alpha_n^2 < \infty$
- Usually we will take $\alpha_n = 1/n$

Rewriting the recursion:

$$\theta(n+1) = \theta(n) + \alpha_{n+1}[\bar{f}(\theta(n)) + \Delta_{n+1}]$$

Interpretation: $\theta^* \equiv$ *stationary point* of the ODE

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

Analysis: Stability of the ODE $\oplus$ (See Borkar's monograph) $\Longrightarrow$

$$\lim_{n \to \infty} \theta(n) = \theta^*$$

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 $\{\Sigma_n\}$

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

$$A = \tfrac{d}{d\theta}\bar{f}\,(\theta^*)$$
$$\Sigma_\Delta = \mathsf{E}[\Delta_{n+1}\Delta_{n+1}^{\intercal}]$$

Asymptotic Variance Theory

1. If $\operatorname{Re}\lambda(A) \geq -\frac{1}{2}$ for some eigenvalue then Σ is (typically) infinite
2. If $\operatorname{Re}\lambda(A) < -\frac{1}{2}$ for all, $\Sigma = \lim_{n\to\infty} \Sigma_n$ solves 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 $\{G_n\}$:

$$\theta(n+1) = \theta(n) + \alpha_{n+1} G_{n+1} f(\theta(n), W(n+1))$$

Optimal Asymptotic Covariance

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

$$\theta(n+1) = \theta(n) + \alpha_{n+1} G_{n+1} f(\theta(n), W(n+1))$$

Assume it converges, and linearize:

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

Optimal Asymptotic Covariance

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

$$\theta(n+1) = \theta(n) + \alpha_{n+1} G_{n+1} f(\theta(n), W(n+1))$$

Assume it converges, and linearize:

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

Asymptotic Variance Theory

1. If $\operatorname{Re} \lambda(GA) \geq -\frac{1}{2}$ for some eigenvalue then Σ^G is (typically) infinite
2. If $\operatorname{Re} \lambda(GA) < -\frac{1}{2}$ for all, Σ^G solves the Lyapunov equation:

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

Optimal Asymptotic Covariance

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

$$\theta(n+1) = \theta(n) + \alpha_{n+1} G_{n+1} f(\theta(n), W(n+1))$$

Assume it converges, and linearize:

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

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

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

Polyak-Ruppert averaging [11] is also optimal, but first two bullets are missing.

Monte Carlo Example

$\alpha_n = \frac{g}{n}$, $g > 0$

Find $\theta^* = E[c(W)]$:

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

Monte Carlo Example

$\alpha_n = \frac{g}{n}$, $g > 0$

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

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

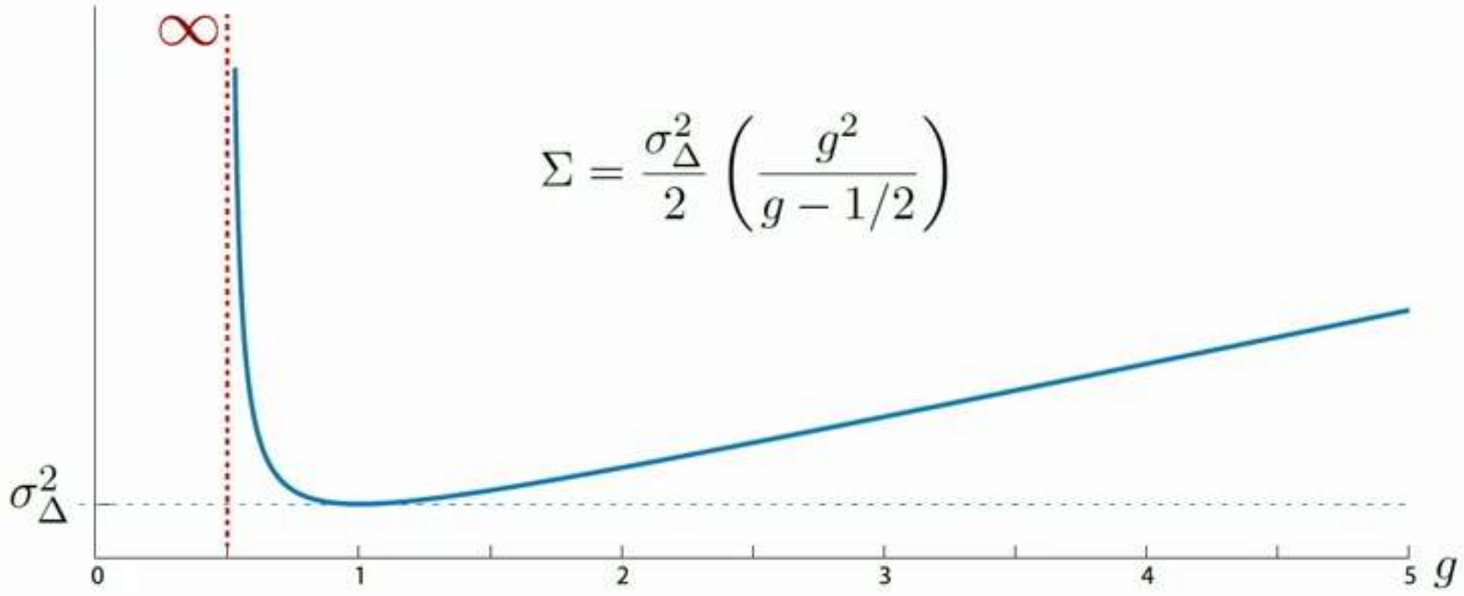
Monte Carlo Example

$\alpha_n = \frac{g}{n}$, $g > 0$

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

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

Example: $c(w) = w^2$, $W \sim N(0,1)$



Asymptotic variance as a function of g

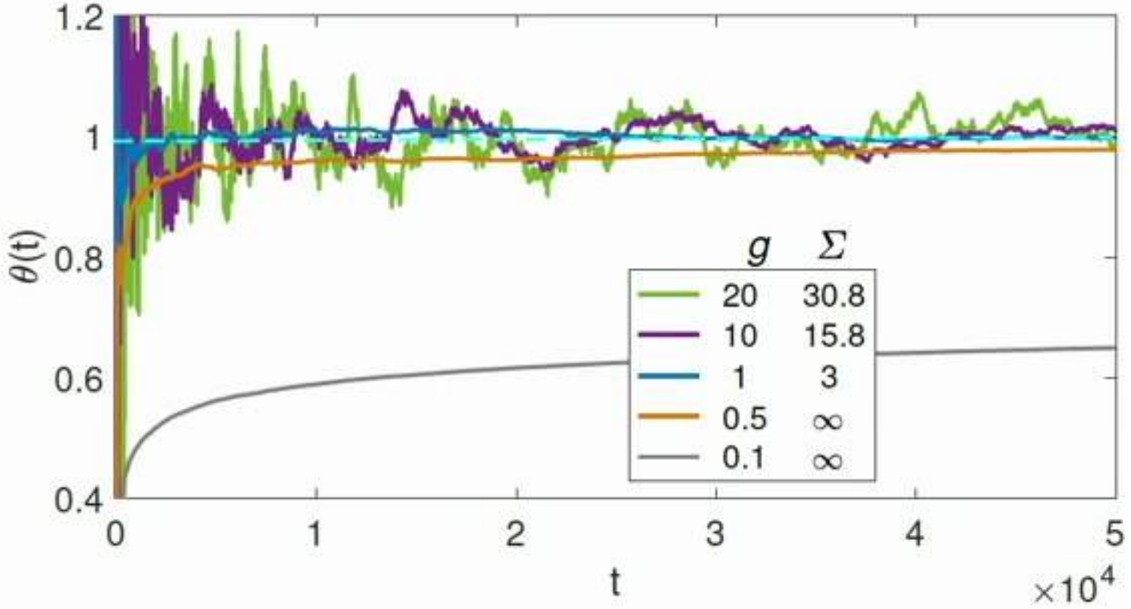
Monte Carlo Example

$\alpha_n = \frac{g}{n}$, $g > 0$

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

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

Example: $c(w) = w^2$, $W \sim N(0,1)$



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

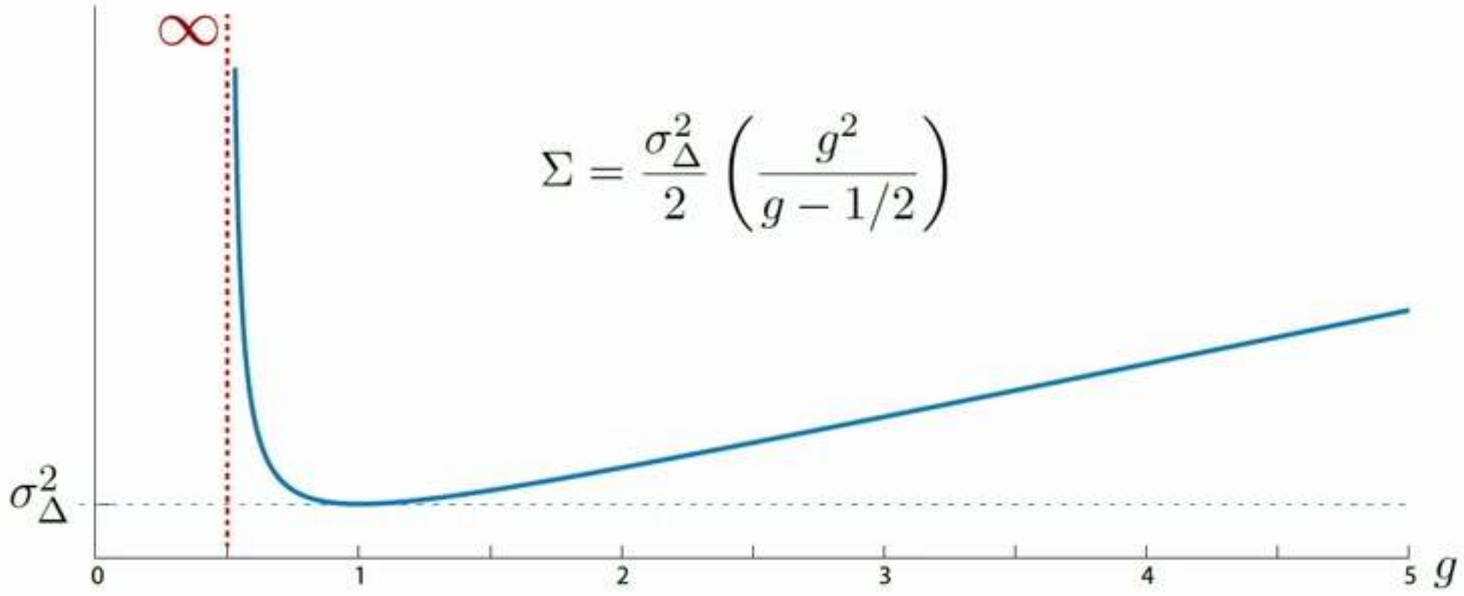
Monte Carlo Example

$\alpha_n = \frac{g}{n}$, $g > 0$

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

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

Example: $c(w) = w^2$, $W \sim N(0,1)$



Asymptotic variance as a function of g

Optimal Variance and SNR

$\bar{f}(\theta) = A\theta - b$ $\qquad$ $\frac{\partial}{\partial\theta}(\bar{f}(\theta)) = A$

Stochastic Newton Raphson: Matrix gain algorithm with $G_n \approx G^* = -A^{-1}$:

SNR Algorithm:

$$\theta(n+1) = \theta(n) + \alpha_{n+1} G_n f(\theta(n), W(n+1))$$

$$G_n^{-1} = -\frac{1}{n+1}\sum_{k=1}^{n+1} A_k \qquad A_{n+1} = \frac{d}{d\theta} f(\theta(n), W(n+1))$$

Optimal Variance and SNR

$\bar{f}(\theta) = A\theta - b$ $\qquad$ $\frac{\partial}{\partial\theta}(\bar{f}(\theta)) = A$

Stochastic Newton Raphson: Matrix gain algorithm with $G_n \approx G^* = -A^{-1}$:

SNR Algorithm:

$$\theta(n+1) = \theta(n) + \alpha_{n+1}(-\widehat{A}_{n+1})^{-1} f(\theta(n), W(n+1))$$

$$\widehat{A}_{n+1} = \widehat{A}_n + \alpha_{n+1}(A_{n+1} - \widehat{A}_n)$$

Optimal Variance and SNR

$\bar{f}(\theta) = A\theta - b \qquad \frac{\partial}{\partial\theta}(\bar{f}(\theta)) = A$

Stochastic Newton Raphson: Matrix gain algorithm with $G_n \approx G^* = -A^{-1}$:

SNR Algorithm:

$$\theta(n+1) = \theta(n) + \alpha_{n+1}(-\widehat{A}_{n+1})^{-1} f(\theta(n), W(n+1))$$
$$\widehat{A}_{n+1} = \widehat{A}_n + \alpha_{n+1}(A_{n+1} - \widehat{A}_n)$$

Example: LSTD(λ), *but this was* not *their motivation!*

Optimal Variance and Zap-SNR

$A(\theta) = \frac{\partial}{\partial\theta}\bar{f}(\theta)$ is a function of θ

Zap-SNR (designed to emulate deterministic Newton-Raphson)

$$\text{Requires} \quad \widehat{A}_{n+1} \approx A(\theta_n) := \frac{d}{d\theta}\bar{f}\,(\theta_n)$$

Optimal Variance and Zap-SNR

$A(\theta) = \frac{\partial}{\partial\theta}\bar{f}(\theta)$ is a function of θ

Zap-SNR (designed to emulate deterministic Newton-Raphson)

$$\theta(n+1) = \theta(n) + \alpha_{n+1}(-\widehat{A}_{n+1})^{-1} f(\theta(n), W(n+1))$$

$$\widehat{A}_{n+1} = \widehat{A}_n + \gamma_{n+1}(A_{n+1} - \widehat{A}_n), \qquad A_{n+1} = \frac{d}{d\theta} f(\theta(n), W(n+1))$$

$\widehat{A}_{n+1} \approx A(\theta_n)$ requires high-gain, $\dfrac{\gamma_n}{\alpha_n} \to \infty, \qquad n \to \infty$

Optimal Variance and Zap-SNR

$A(\theta) = \frac{\partial}{\partial\theta}\bar{f}(\theta)$ is a function of θ

Zap-SNR (designed to emulate deterministic Newton-Raphson)

$$\theta(n+1) = \theta(n) + \alpha_{n+1}(-\widehat{A}_{n+1})^{-1} f(\theta(n), W(n+1))$$

$$\widehat{A}_{n+1} = \widehat{A}_n + \gamma_{n+1}(A_{n+1} - \widehat{A}_n), \qquad A_{n+1} = \frac{d}{d\theta} f(\theta(n), W(n+1))$$

$\widehat{A}_{n+1} \approx A(\theta_n)$ requires high-gain, $\dfrac{\gamma_n}{\alpha_n} \to \infty, \qquad n \to \infty$

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

Optimal Variance and Zap-SNR

$A(\theta) = \frac{\partial}{\partial\theta}\bar{f}(\theta)$ is a function of θ

Zap-SNR (designed to emulate deterministic Newton-Raphson)

$$\theta(n+1) = \theta(n) + \alpha_{n+1}(-\widehat{A}_{n+1})^{-1} f(\theta(n), W(n+1))$$

$$\widehat{A}_{n+1} = \widehat{A}_n + \gamma_{n+1}(A_{n+1} - \widehat{A}_n), \qquad A_{n+1} = \frac{d}{d\theta} f(\theta(n), W(n+1))$$

$\widehat{A}_{n+1} \approx A(\theta_n)$ requires high-gain, $\dfrac{\gamma_n}{\alpha_n} \to \infty, \qquad n \to \infty$

ODE for Zap-SNR

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

Optimal Variance and Zap-SNR

$A(\theta) = \frac{\partial}{\partial\theta}\bar{f}(\theta)$ is a function of θ

Zap-SNR (designed to emulate deterministic Newton-Raphson)

$$\theta(n+1) = \theta(n) + \alpha_{n+1}(-\widehat{A}_{n+1})^{-1} f(\theta(n), W(n+1))$$

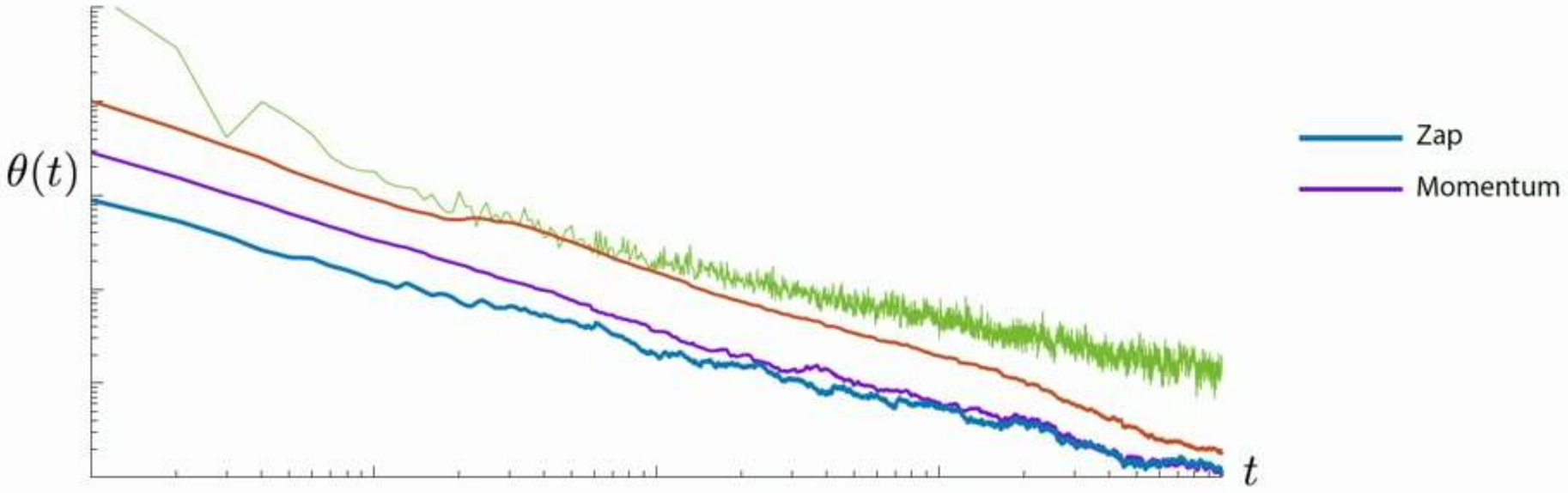
$$\widehat{A}_{n+1} = \widehat{A}_n + \gamma_{n+1}(A_{n+1} - \widehat{A}_n), \qquad A_{n+1} = \frac{d}{d\theta} f(\theta(n), W(n+1))$$

$\widehat{A}_{n+1} \approx A(\theta_n)$ requires high-gain, $\dfrac{\gamma_n}{\alpha_n} \to \infty, \qquad n \to \infty$

ODE for Zap-SNR

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

- Not necessarily stable (just like in deterministic Newton-Raphson)
- *General* conditions for convergence is open



Optimal Momentum
Stochastic Approximation

Momentum based Stochastic Approximation

$\Delta\theta(n) := \theta(n) - \theta(n-1)$ $\quad$ $f_{n+1}(\theta) := f(\theta, W_{n+1})$

Matrix Gain Stochastic Approximation:

$$\Delta\theta(n+1) = \alpha_n G_{n+1} f_{n+1}(\theta(n))$$

Heavy Ball Stochastic Approximation:

following Polyak, 1964 [12]

$$\Delta\theta(n+1) = m\Delta\theta(n) + \alpha_n f_{n+1}(\theta(n))$$

Matirx Heavy Ball Stochastic Approximation:

$$\Delta\theta(n+1) = M_{n+1}\Delta\theta(n) + \alpha_n G_{n+1} f_{n+1}(\theta(n))$$

Matrix Heavy Ball Stochastic Approximation

Optimizing $\{M_{n+1}\}$ and $\{G_{n+1}\}$

Matirx Heavy Ball Stochastic Approximation:

$$\Delta\theta(n+1) = M_{n+1}\Delta\theta(n) + \alpha_{n+1}G_{n+1}f_{n+1}(\theta(n))$$

Heuristic: Assume $\Delta\theta_n \to 0$ much faster than $\theta_n \to \theta^*$

Matrix Heavy Ball Stochastic Approximation

Optimizing $\{M_{n+1}\}$ and $\{G_{n+1}\}$

Matirx Heavy Ball Stochastic Approximation:

$$\Delta\theta(n+1) = M_{n+1}\Delta\theta(n) + \alpha_{n+1}G_{n+1}f_{n+1}(\theta(n))$$

Heuristic: Assume $\Delta\theta_n \to 0$ much faster than $\theta_n \to \theta^*$

$$\Delta\theta(n+1) \approx M_{n+1}\Delta\theta(n+1) + \alpha_{n+1}G_{n+1}f_{n+1}(\theta(n))$$

$$\approx [I - M_{n+1}]^{-1}\alpha_{n+1}G_{n+1}f_{n+1}(\theta(n))$$

Momentum based Stochastic Approximation

$\Delta\theta(n) := \theta(n) - \theta(n-1)$ $\quad$ $f_{n+1}(\theta) := f(\theta, W_{n+1})$

Matrix Gain Stochastic Approximation:

$$\Delta\theta(n+1) = \alpha_n G_{n+1} f_{n+1}(\theta(n))$$

Heavy Ball Stochastic Approximation:

following Polyak, 1964 [12]

$$\Delta\theta(n+1) = m\Delta\theta(n) + \alpha_n f_{n+1}(\theta(n))$$

Matirx Heavy Ball Stochastic Approximation:

$$\Delta\theta(n+1) = M_{n+1}\Delta\theta(n) + \alpha_n G_{n+1} f_{n+1}(\theta(n))$$

Matrix Heavy Ball Stochastic Approximation

Optimizing $\{M_{n+1}\}$ and $\{G_{n+1}\}$

Matirx Heavy Ball Stochastic Approximation:

$$\Delta\theta(n+1) = M_{n+1}\Delta\theta(n) + \alpha_{n+1}G_{n+1}f_{n+1}(\theta(n))$$

Heuristic: Assume $\Delta\theta_n \to 0$ much faster than $\theta_n \to \theta^*$

$$\Delta\theta(n+1) \approx M_{n+1}\Delta\theta(n+1) + \alpha_{n+1}G_{n+1}f_{n+1}(\theta(n))$$

$$\approx [I - M_{n+1}]^{-1}\alpha_{n+1}G_{n+1}f_{n+1}(\theta(n))$$

Try this: $M_{n+1} = I + G_{n+1}\widehat{A}_{n+1}$

Momentum based Stochastic Approximation

$\widehat{A}_{n+1} \approx \frac{d}{d\theta} \bar{f}(\theta_n)$

SNR: $\Delta\theta(n+1) = -\alpha_{n+1} \widehat{A}_{n+1}^{-1} f_{n+1}(\theta(n))$

Momentum based Stochastic Approximation

$\widehat{A}_{n+1} \approx \frac{d}{d\theta} \bar{f}\,(\theta_n)$

Special case $G_n = \zeta I$

$$\text{SNR:} \quad \Delta\theta(n+1) = -\alpha_{n+1} \widehat{A}_{n+1}^{-1} f_{n+1}(\theta(n))$$

$$\text{PolSA:} \quad \Delta\theta(n+1) = [I + \zeta \widehat{A}_{n+1}]\Delta\theta(n) + \alpha_{n+1}\zeta f_{n+1}(\theta(n))$$

Matrix Heavy Ball Stochastic Approximation

Optimizing $\{M_{n+1}\}$ and $\{G_{n+1}\}$

Matirx Heavy Ball Stochastic Approximation:

$$\Delta\theta(n+1) = M_{n+1}\Delta\theta(n) + \alpha_{n+1}G_{n+1}f_{n+1}(\theta(n))$$

Heuristic: Assume $\Delta\theta_n \to 0$ much faster than $\theta_n \to \theta^*$

$$\Delta\theta(n+1) \approx M_{n+1}\Delta\theta(n+1) + \alpha_{n+1}G_{n+1}f_{n+1}(\theta(n))$$

$$\approx [I - M_{n+1}]^{-1}\alpha_{n+1}G_{n+1}f_{n+1}(\theta(n))$$

Try this: $M_{n+1} = I + G_{n+1}\widehat{A}_{n+1}$

$$= -\alpha_{n+1}\widehat{A}_{n+1}^{-1}f_{n+1}(\theta(n)) \qquad \text{SNR!}$$

Momentum based Stochastic Approximation

$\widehat{A}_{n+1} \approx \frac{d}{d\theta} \bar{f}\,(\theta_n)$

Special case $G_n = \zeta I$

$$\text{SNR:} \quad \Delta\theta(n+1) = -\alpha_{n+1} \widehat{A}_{n+1}^{-1} f_{n+1}(\theta(n))$$

$$\text{PolSA:} \quad \Delta\theta(n+1) = [I + \zeta \widehat{A}_{n+1}]\Delta\theta(n) + \alpha_{n+1}\zeta f_{n+1}(\theta(n))$$

Momentum based Stochastic Approximation

$\widehat{A}_{n+1} \approx \frac{d}{d\theta} \bar{f}\,(\theta_n)$

Special case $G_n = \zeta I$

SNR: $\Delta\theta(n+1) = -\alpha_{n+1}\widehat{A}_{n+1}^{-1} f_{n+1}(\theta(n))$

PolSA: $\Delta\theta(n+1) = [I + \zeta\widehat{A}_{n+1}]\Delta\theta(n) + \alpha_{n+1}\zeta f_{n+1}(\theta(n))$

NeSA: $\Delta\theta(n+1) = \Delta\theta(n) + \zeta[f_{n+1}(\theta(n)) - f_{n+1}(\theta(n-1))] + \alpha_{n+1}\zeta f_{n+1}(\theta(n))$

following Nesterov, 1983 [13]

Momentum based Stochastic Approximation

$\widehat{A}_{n+1} \approx \frac{d}{d\theta} \bar{f}\,(\theta_n)$

Special case $G_n = \zeta I$

$$\text{SNR:} \quad \Delta\theta(n+1) = -\alpha_{n+1} \widehat{A}_{n+1}^{-1} f_{n+1}(\theta(n))$$

$$\text{PolSA:} \quad \Delta\theta(n+1) = [I + \zeta \widehat{A}_{n+1}]\Delta\theta(n) + \alpha_{n+1}\zeta f_{n+1}(\theta(n))$$

$$\text{(linearized) NeSA:} \quad \Delta\theta(n+1) = [I + \zeta A_{n+1}]\Delta\theta(n) + \alpha_{n+1}\zeta f_{n+1}(\theta(n))$$

Momentum based Stochastic Approximation

$\widehat{A}_{n+1} \approx \frac{d}{d\theta} \bar{f}\,(\theta_n)$

Special case $G_n = \zeta I$

SNR: $\Delta\theta(n+1) = -\alpha_{n+1}\widehat{A}_{n+1}^{-1} f_{n+1}(\theta(n))$

PolSA: $\Delta\theta(n+1) = [I + \zeta\widehat{A}_{n+1}]\Delta\theta(n) + \alpha_{n+1}\zeta f_{n+1}(\theta(n))$

NeSA: $\Delta\theta(n+1) = \Delta\theta(n) + \zeta[f_{n+1}(\theta(n)) - f_{n+1}(\theta(n-1))]$
$+ \alpha_{n+1}\zeta f_{n+1}(\theta(n))$

following Nesterov, 1983 [13]

Momentum based Stochastic Approximation

$\widehat{A}_{n+1} \approx \frac{d}{d\theta} \bar{f}\,(\theta_n)$

Special case $G_n = \zeta I$

$$\text{SNR:} \quad \Delta\theta(n+1) = -\alpha_{n+1} \widehat{A}_{n+1}^{-1} f_{n+1}(\theta(n))$$

$$\text{PolSA:} \quad \Delta\theta(n+1) = [I + \zeta \widehat{A}_{n+1}]\Delta\theta(n) + \alpha_{n+1}\zeta f_{n+1}(\theta(n))$$

$$\text{(linearized) NeSA:} \quad \Delta\theta(n+1) = [I + \zeta A_{n+1}]\Delta\theta(n) + \alpha_{n+1}\zeta f_{n+1}(\theta(n))$$

Momentum based Stochastic Approximation

$\widehat{A}_{n+1} \approx \frac{d}{d\theta} \bar{f}\,(\theta_n)$

Special case $G_n = \zeta I$

$$\text{SNR:} \quad \Delta\theta(n+1) = -\alpha_{n+1} \widehat{A}_{n+1}^{-1} f_{n+1}(\theta(n))$$

$$\text{PolSA:} \quad \Delta\theta(n+1) = [I + \zeta \widehat{A}_{n+1}]\Delta\theta(n) + \alpha_{n+1}\zeta f_{n+1}(\theta(n))$$

$$\text{(linearized) NeSA:} \quad \Delta\theta(n+1) = [I + \zeta A_{n+1}]\Delta\theta(n) + \alpha_{n+1}\zeta f_{n+1}(\theta(n))$$

Coupling of SNR and PolSA: $\|\theta_n^{\text{SNR}} - \theta_n^{\text{PolSA}}\| = O(n^{-1})$

- Linear model: $f_{n+1}(\theta) = A(\theta - \theta^*) + \Delta_{n+1}$
- $\{\Delta_{n+1}\}$ square-integrable martingale difference sequence
- A Hurwitz *and* $\text{eig}(I + \zeta A) \in$ open unit disk

PolSA has optimal asymptotic variance

Momentum based Stochastic Approximation

$\widehat{A}_{n+1} \approx \frac{d}{d\theta} \bar{f}\,(\theta_n)$

Special case $G_n = \zeta I$

$$\text{SNR:} \quad \Delta\theta(n+1) = -\alpha_{n+1} \widehat{A}_{n+1}^{-1} f_{n+1}(\theta(n))$$

$$\text{PolSA:} \quad \Delta\theta(n+1) = [I + \zeta \widehat{A}_{n+1}]\Delta\theta(n) + \alpha_{n+1}\zeta f_{n+1}(\theta(n))$$

$$\text{(linearized) NeSA:} \quad \Delta\theta(n+1) = [I + \zeta A_{n+1}]\Delta\theta(n) + \alpha_{n+1}\zeta f_{n+1}(\theta(n))$$

Momentum based Stochastic Approximation

$\widehat{A}_{n+1} \approx \frac{d}{d\theta} \bar{f}\,(\theta_n)$

Special case $G_n = \zeta I$

$$\text{SNR:} \quad \Delta\theta(n+1) = -\alpha_{n+1} \widehat{A}_{n+1}^{-1} f_{n+1}(\theta(n))$$

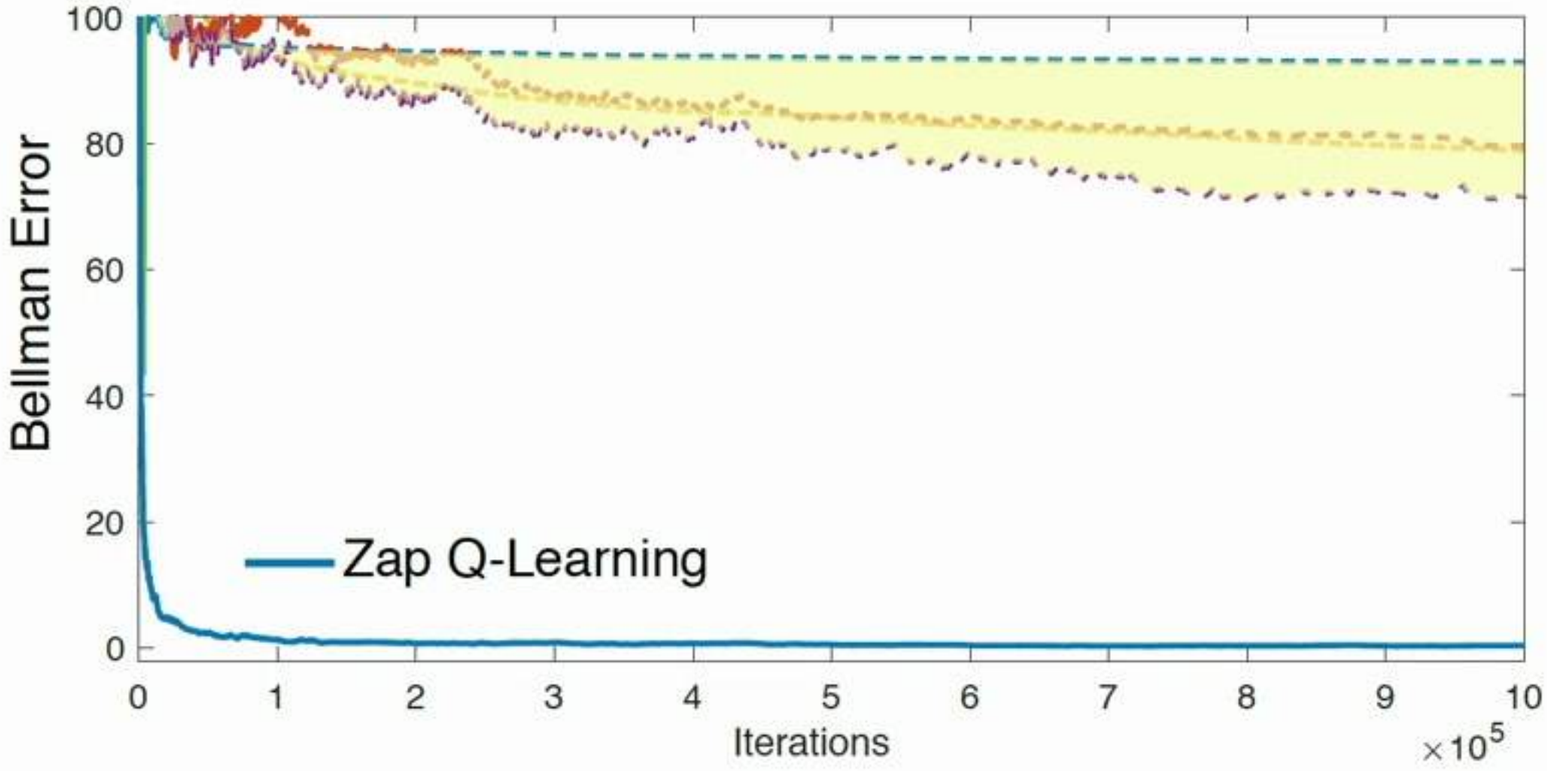
$$\text{PolSA:} \quad \Delta\theta(n+1) = [I + \zeta \widehat{A}_{n+1}]\Delta\theta(n) + \alpha_{n+1}\zeta f_{n+1}(\theta(n))$$

$$\text{(linearized) NeSA:} \quad \Delta\theta(n+1) = [I + \zeta A_{n+1}]\Delta\theta(n) + \alpha_{n+1}\zeta f_{n+1}(\theta(n))$$

Coupling of SNR and PolSA: $\|\theta_n^{\text{SNR}} - \theta_n^{\text{PolSA}}\| = O(n^{-1})$

- Linear model: $f_{n+1}(\theta) = A(\theta - \theta^*) + \Delta_{n+1}$
- $\{\Delta_{n+1}\}$ square-integrable martingale difference sequence
- A Hurwitz *and* $\text{eig}(I + \zeta A) \in$ open unit disk

PolSA has optimal asymptotic variance



Reinforcement Learning and Stochastic Approximation

Stochastic Optimal Control

MDP Model

$\boldsymbol{X}$ is a stationary 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: $\mathsf{X} \times \mathsf{U} \to \mathbb{R}$ is a cost function
- $\beta < 1$ a discount factor

Stochastic Optimal Control

MDP Model

$\boldsymbol{X}$ is a stationary 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: $\mathsf{X} \times \mathsf{U} \to \mathbb{R}$ is a cost function
- $\beta < 1$ a discount factor

Q function:

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

Bellman equation:

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

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

Given $\{Q^\theta : \theta \in \mathbb{R}^d\}$, find θ^* that solves

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

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

Stochastic Optimal Control

MDP Model

$\boldsymbol{X}$ is a stationary 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: $\mathsf{X} \times \mathsf{U} \to \mathbb{R}$ is a cost function
- $\beta < 1$ a discount factor

Q function:

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

Bellman equation:

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

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

Given $\{Q^\theta : \theta \in \mathbb{R}^d\}$, find θ^* that solves

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

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

Stochastic Optimal Control

MDP Model

$\boldsymbol{X}$ is a stationary 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: $\mathsf{X} \times \mathsf{U} \to \mathbb{R}$ is a cost function
- $\beta < 1$ a discount factor

Q function:

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

Bellman equation:

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

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

Given $\{Q^\theta : \theta \in \mathbb{R}^d\}$, find θ^* that solves

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

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

Q-learning and Stochastic Approximation

Q-learning: $Q^\theta(x,u) = \theta^\intercal \psi(x,u)$ $\qquad \theta \in \mathbb{R}^d, \quad \psi : \mathsf{X} \times \mathsf{U} \to \mathbb{R}^d$

Find θ^* such that:

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

Example: $\zeta_n \equiv \psi(X(n),U(n))$

Q-learning and Stochastic Approximation

Q-learning: $Q^\theta(x,u) = \theta^\intercal \psi(x,u)$ $\qquad \theta \in \mathbb{R}^d, \quad \psi : \mathsf{X} \times \mathsf{U} \to \mathbb{R}^d$

Find θ^* such that:

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

Example: $\zeta_n \equiv \psi(X(n),U(n))$

Q-learning and SA

Root finding problem: $\qquad \bar{f}(\theta^*) = \boldsymbol{A}(\theta^*)\theta^* - \boldsymbol{b} = 0$

$$\boldsymbol{A}(\theta) = \mathsf{E}\big[\zeta_n\big[\beta\psi(X(n+1),\phi^\theta(X(n+1))) - \psi(X(n),U(n))\big]^\intercal\big]$$

$$\boldsymbol{b} := \mathsf{E}\big[\zeta_n c(X_n,U_n)\big]$$

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

Watkins' Q-learning

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

Watkins' Q-learning

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

Watkin's algorithm is Stochastic Approximation

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

- Linearly parameterized family of functions: $Q^\theta(x,u) = \theta^\intercal \psi(x,u)$
- $\zeta_n \equiv \psi(X(n),U(n))$
- $\psi_i(x,u) = 1\{x = x^i, u = u^i\}$ (complete basis)

Watkins' Q-learning

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

Watkin's algorithm is Stochastic Approximation

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

- Linearly parameterized family of functions: $Q^\theta(x,u) = \theta^\intercal \psi(x,u)$
- $\zeta_n \equiv \psi(X(n),U(n))$
- $\psi_i(x,u) = 1\{x = x^i, u = u^i\}$ (complete basis)

Converges, but has infinite asymptotic variance if $\beta > \frac{1}{2}$:

$$\lambda_{\max}\big(\boldsymbol{A}(\theta^*)\big) < 0\,, \quad \lambda_{\max}\big(\boldsymbol{A}(\theta^*)\big) > -\tfrac{1}{2}$$

[D & Meyn, 2017]

Watkins' Q-learning

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

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

Watkin's algorithm is Stochastic Approximation

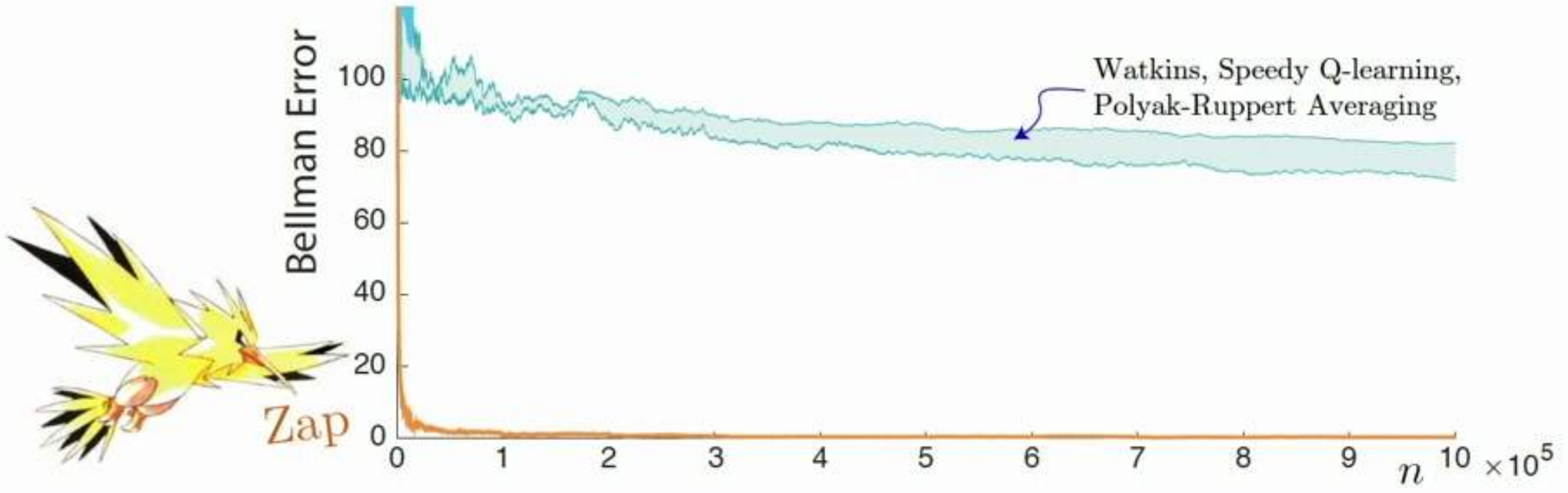
The family $\{Q^\theta\}$ and *eligibility vectors* $\{\zeta_n\}$ in this design:

- Linearly parameterized family of functions: $Q^\theta(x,u) = \theta^\intercal \psi(x,u)$
- $\zeta_n \equiv \psi(X(n),U(n))$
- $\psi_i(x,u) = 1\{x = x^i, u = u^i\}$ (complete basis)

Converges, but has infinite asymptotic variance if $\beta > \frac{1}{2}$:

$$\lambda_{\max}\big(\boldsymbol{A}(\theta^*)\big) < 0\,, \quad \lambda_{\max}\big(\boldsymbol{A}(\theta^*)\big) > -\tfrac{1}{2}$$

[D & Meyn, 2017]



Zap Q-Learning

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:

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

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:

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

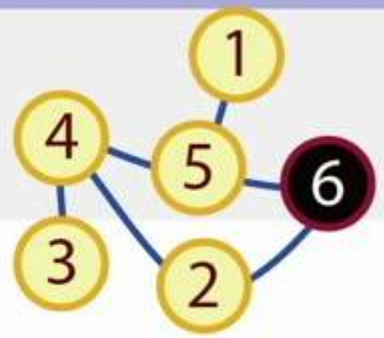
ODE for Zap-Q

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

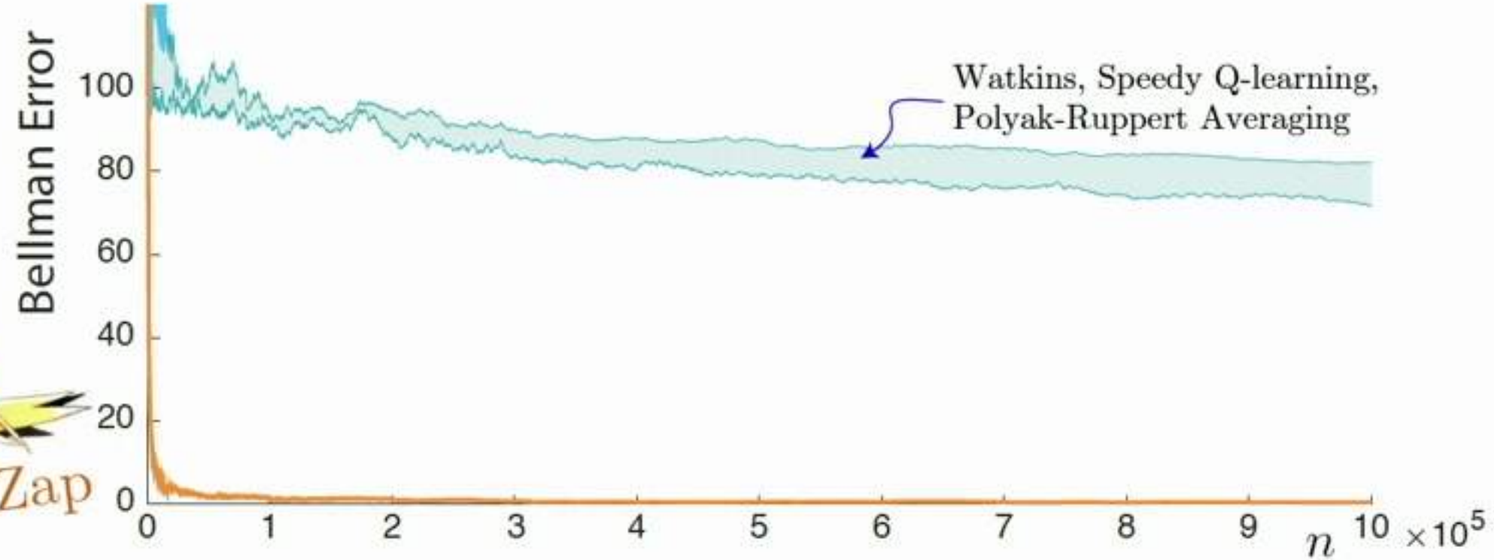
$\Rightarrow$ convergence, optimal covariance, ...

Zap Q-Learning

Example: Stochastic Shortest Path



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

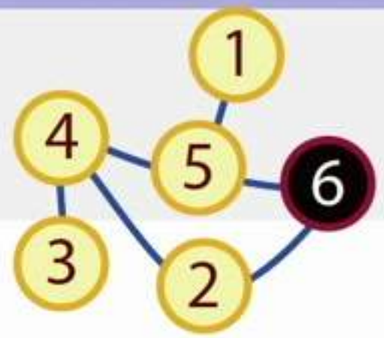


Convergence of Zap-Q Learning

Discount factor: $\beta = 0.99$

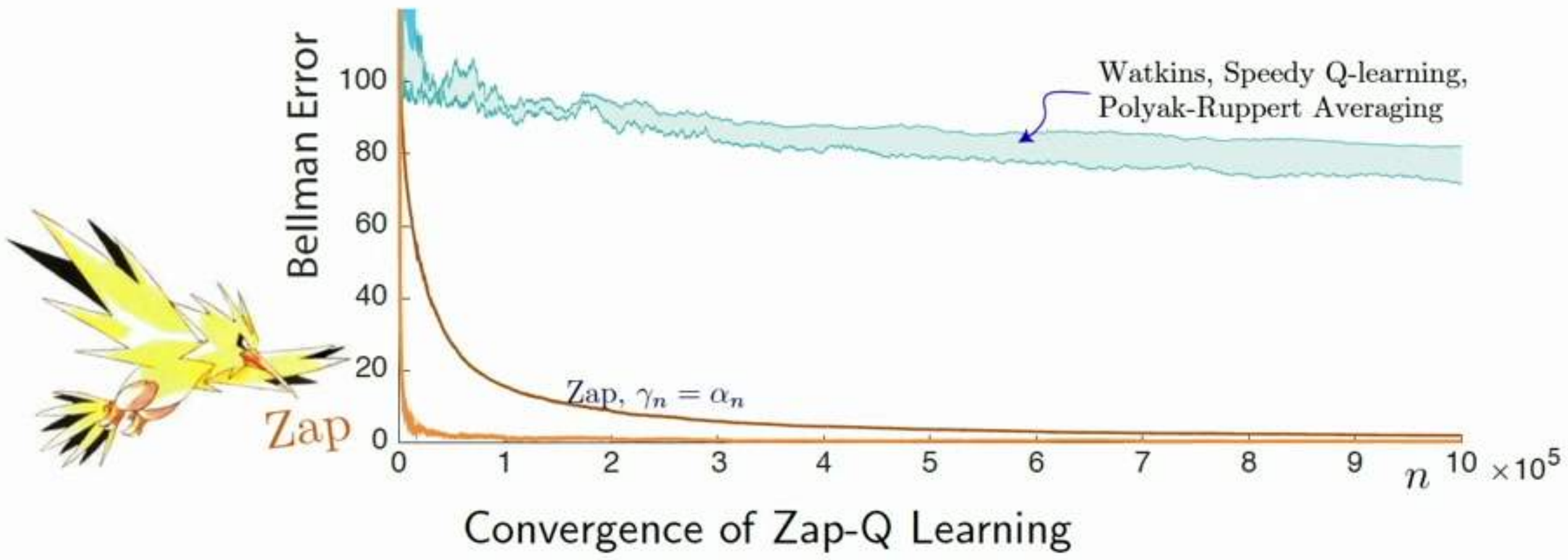
Zap Q-Learning

Example: Stochastic Shortest Path



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

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

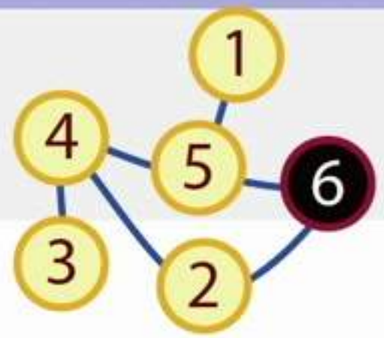


Convergence of Zap-Q Learning

Discount factor: $\beta = 0.99$

Zap Q-Learning

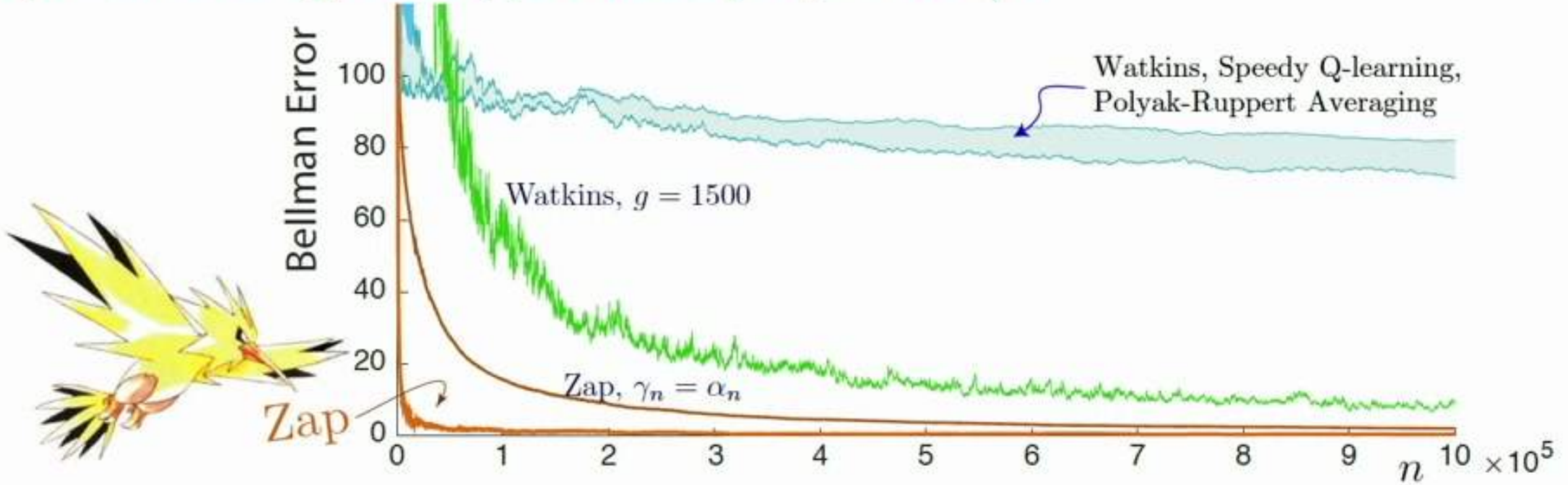
Example: Stochastic Shortest Path



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

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: $\beta = 0.99$

Q-learning and Stochastic Approximation

Q-learning: $Q^\theta(x,u) = \theta^\intercal \psi(x,u) \qquad \theta \in \mathbb{R}^d, \quad \psi : \mathsf{X} \times \mathsf{U} \to \mathbb{R}^d$

Find θ^* such that:

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

Example: $\zeta_n \equiv \psi(X(n),U(n))$

Q-learning and SA

Root finding problem: $\bar{f}(\theta^*) = \boldsymbol{A}(\theta^*)\theta^* - \boldsymbol{b} = 0$

$$\boldsymbol{A}(\theta) = \mathsf{E}\big[\zeta_n\big[\beta\psi(X(n+1),\phi^\theta(X(n+1))) - \psi(X(n),U(n))\big]^\intercal\big]$$
$$\boldsymbol{b} := \mathsf{E}\big[\zeta_n c(X_n,U_n)\big]$$
$$\phi^\theta(x) := \arg\min_u Q^\theta(x,u)$$

Watkins' Q-learning

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

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

Watkin's algorithm is Stochastic Approximation

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

- Linearly parameterized family of functions: $Q^\theta(x,u) = \theta^\intercal \psi(x,u)$
- $\zeta_n \equiv \psi(X(n),U(n))$
- $\psi_i(x,u) = 1\{x = x^i, u = u^i\}$ (complete basis)

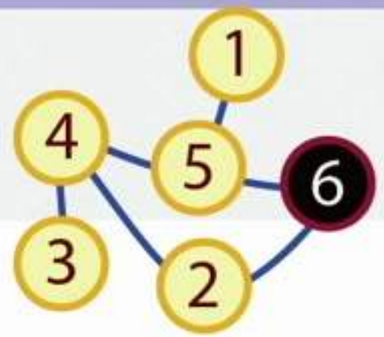
Converges, but has infinite asymptotic variance if $\beta > \frac{1}{2}$:

$$\lambda_{\max}\big(\boldsymbol{A}(\theta^*)\big) < 0\,, \quad \lambda_{\max}\big(\boldsymbol{A}(\theta^*)\big) > -\tfrac{1}{2}$$

[D & Meyn, 2017]

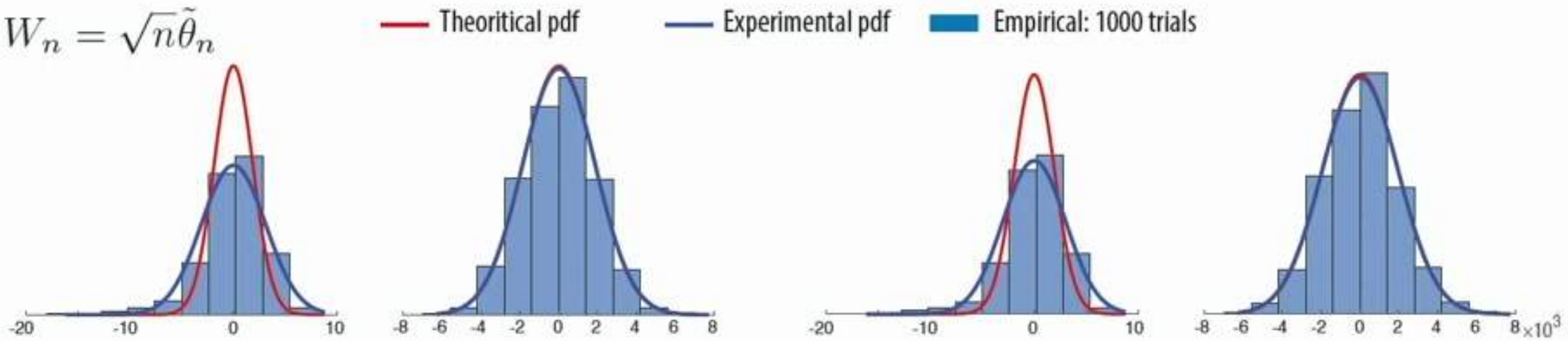
Zap Q-Learning

Optimize Walk to Cafe



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

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



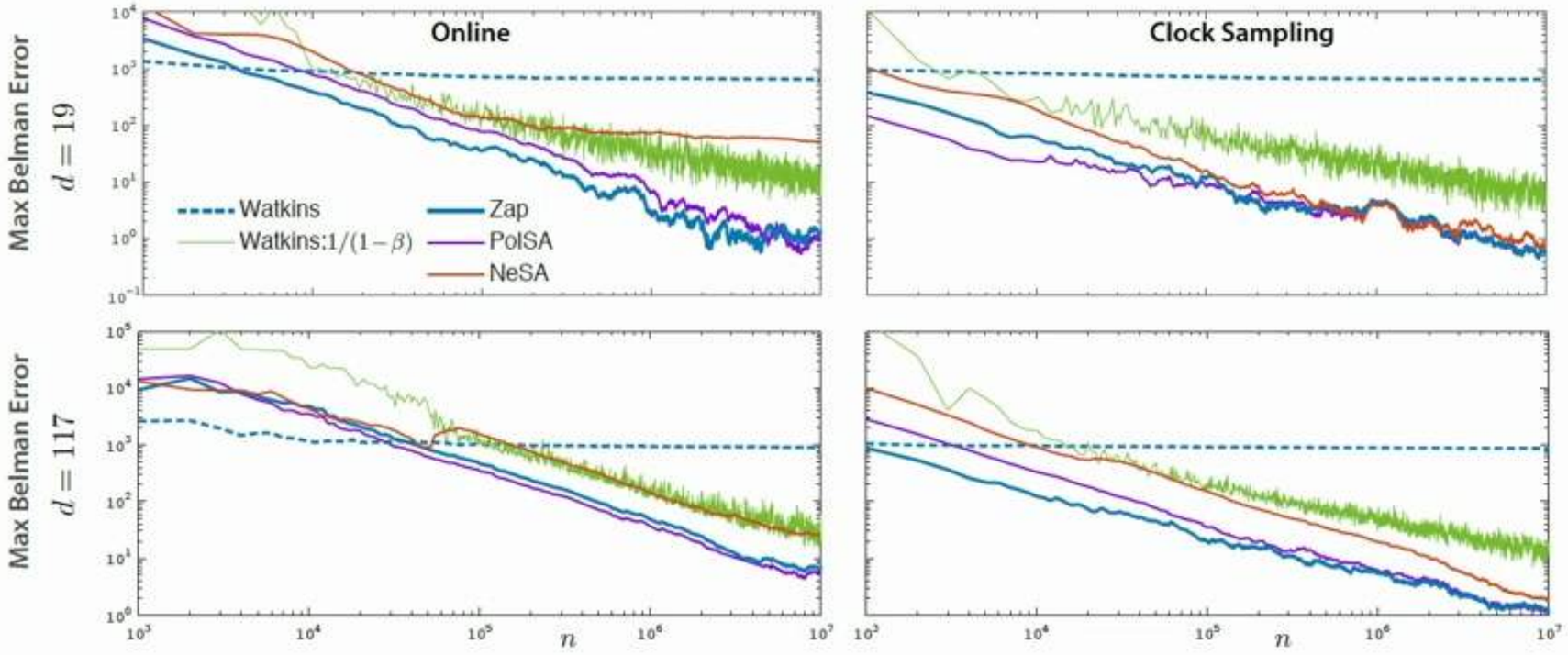
Entry #18: $n = 10^4$ $n = 10^6$ Entry #10: $n = 10^4$ $n = 10^6$

CLT gives good prediction of finite-n performance

Discount factor: $\beta = 0.99$

Zap Q-Learning and Momentum

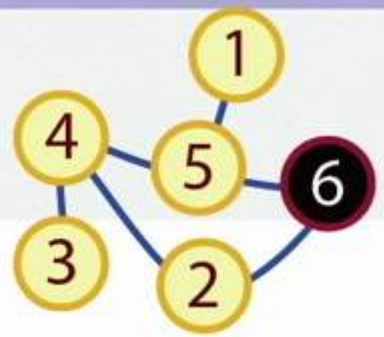
Coupling and convergence for larger models:



Coupling is amazing

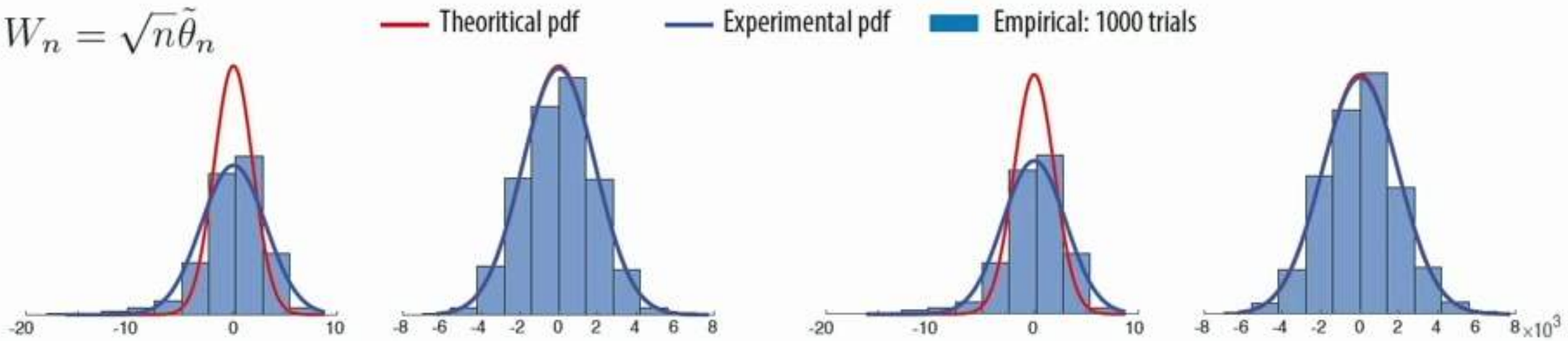
Zap Q-Learning

Optimize Walk to Cafe



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

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



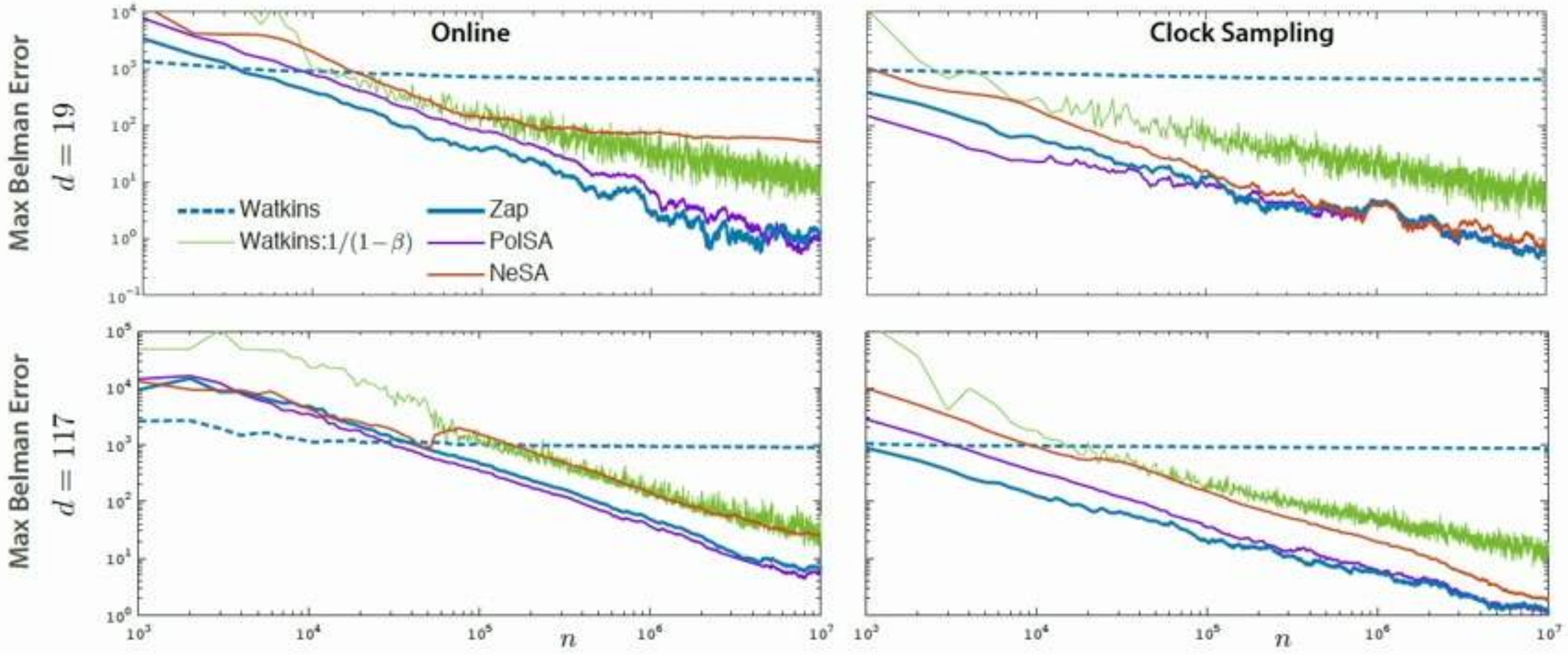
Entry #18: $n = 10^4$ $n = 10^6$ Entry #10: $n = 10^4$ $n = 10^6$

CLT gives good prediction of finite-n performance

Discount factor: $\beta = 0.99$

Zap Q-Learning and Momentum

Coupling and convergence for larger models:



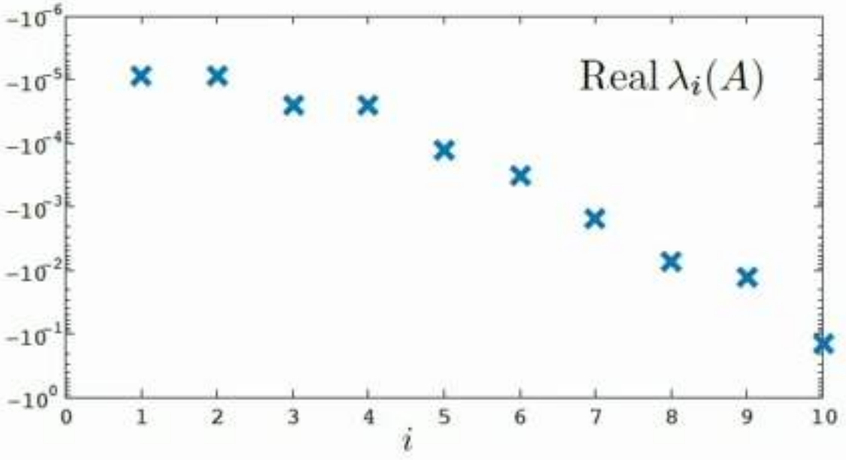
Coupling is amazing

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 $\theta \in \mathbb{R}^{10}$



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

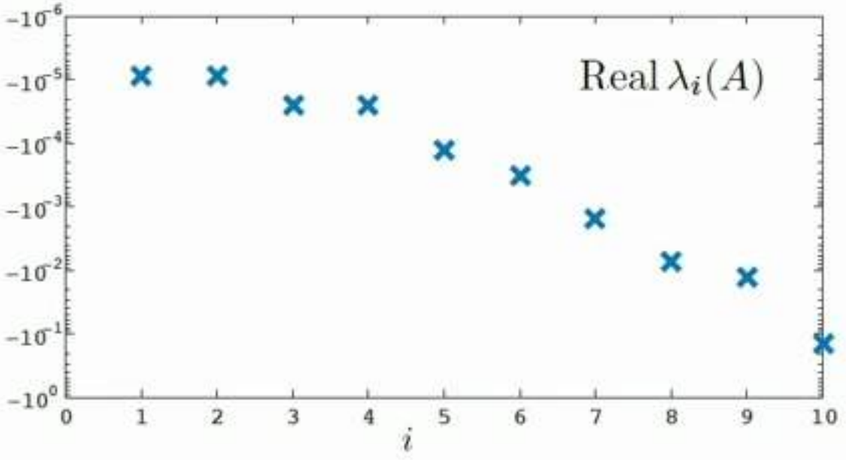
Asymptotic covariance is infinite

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 $\theta \in \mathbb{R}^{10}$



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

Asymptotic covariance is infinite

Authors observed slow convergence
Proposed a matrix gain sequence

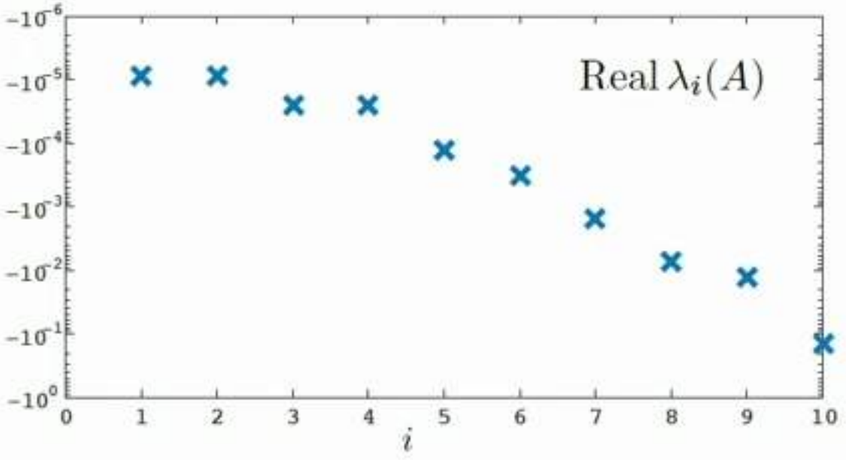
$\{G_n\}$ (see refs for details)

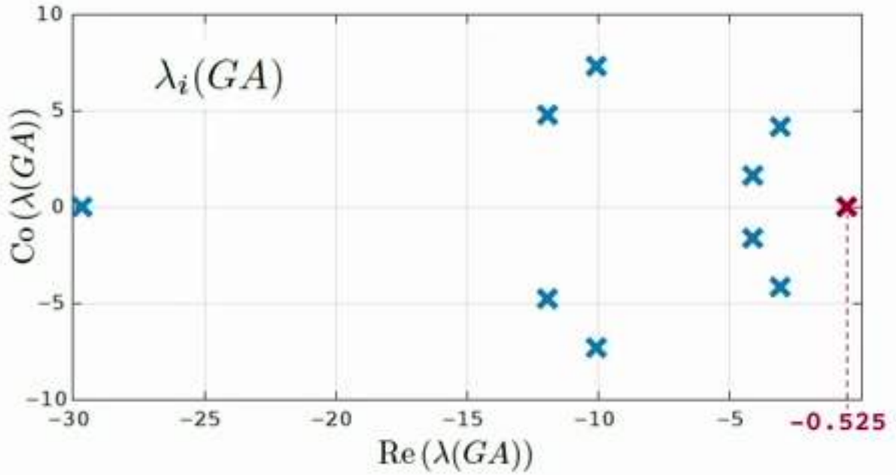
Zap Q-Learning

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 $\text{Re}(\lambda(GA)) < -\frac{1}{2}$

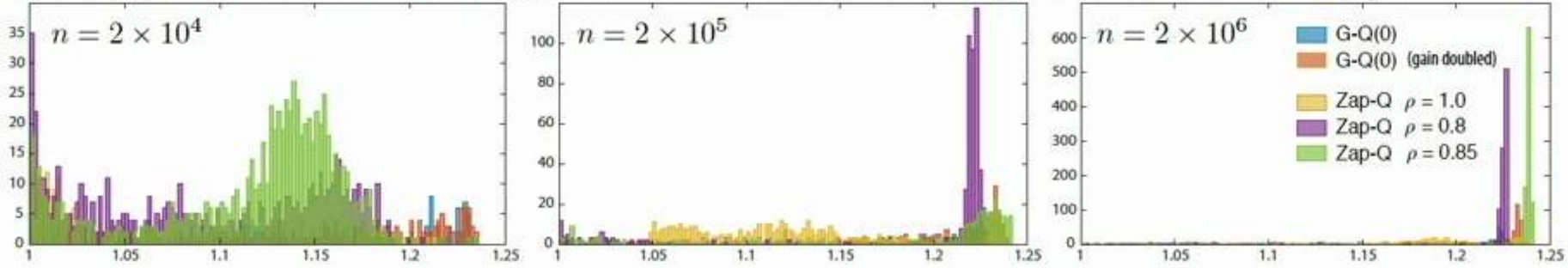
Zap Q-Learning

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

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

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

Histograms of the average reward obtained using the different algorithms:



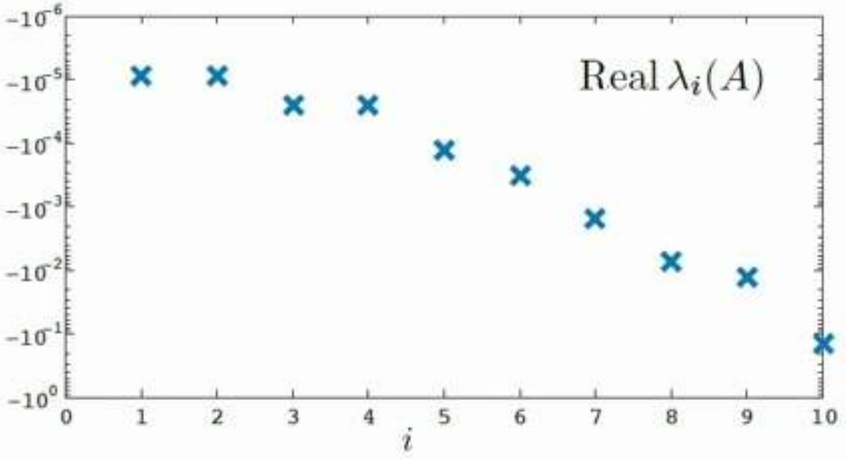
Zap-Q $\gg$ G-Q

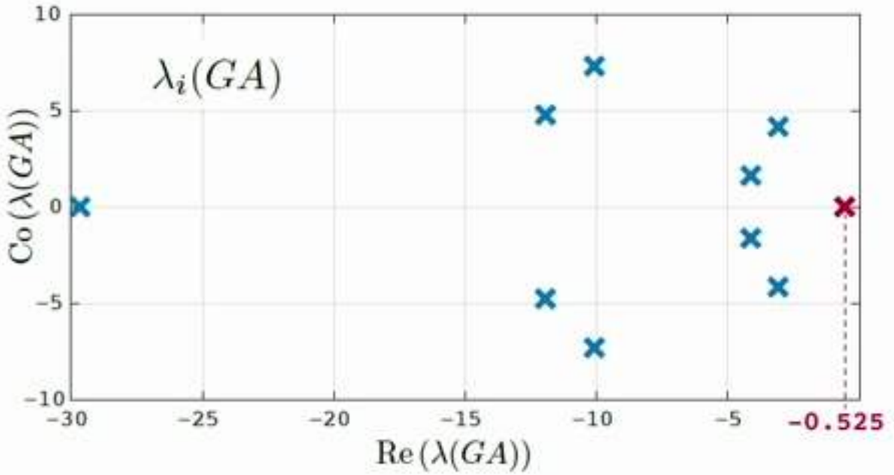
Zap Q-Learning

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 $\text{Re}(\lambda(GA)) < -\frac{1}{2}$

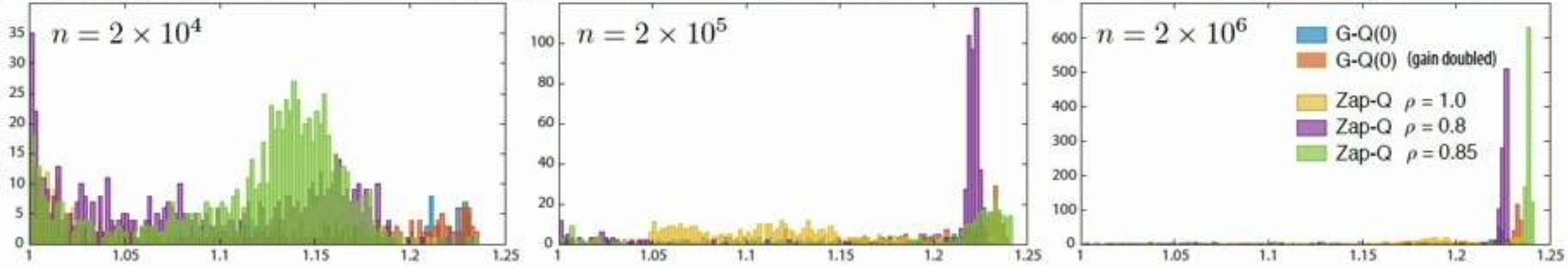
Zap Q-Learning

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

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

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

Histograms of the average reward obtained using the different algorithms:



Zap-Q $\gg$ G-Q

Conclusions & Future Work

Conclusions

- Reinforcement Learning is not just cursed by dimension, but also by variance

  We need better design tools to improve performance

- 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

- PolSA and NeSA prove to be amazing alternatives to the more complex SNR

Conclusions & Future Work

Future Work

- Q-learning with function-approximation
  - *Obtain conditions for a stable algorithm in a general setting*
  - *Optimal stopping time problems (✓)*
- Adaptive optimization of algorithm parameters: critical for momentum methods
- Further reduction in variance using control variates
- Applications in Stochastic Optimization?


Thank you!

This lecture

- A. M. Devraj and S. P. Meyn, *Zap Q-learning. Advances in Neural Information Processing Systems (NIPS).* Dec. 2017.
- A. M. Devraj and S. P. Meyn, *Fastest convergence for Q-learning.* Available on *ArXiv.* Jul. 2017.
- A. M. Devraj, A. Bušić, and S. Meyn. *Zap Meets Momentum: Stochastic Approximation Algorithms with Optimal Convergence Rate.* *ArXiv e-prints*, Sept. 2018.

Berkeley short course, March 2018

- Part I (Basics, with focus on variance of algorithms)

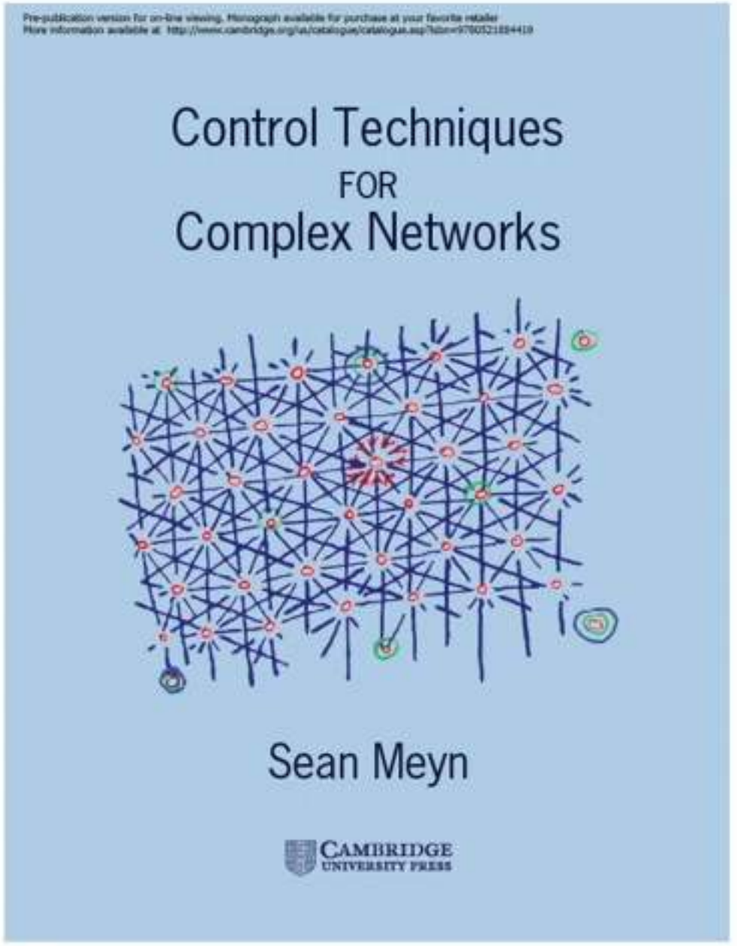
  `https://www.youtube.com/watch?v=dhEF5pfYmvc`

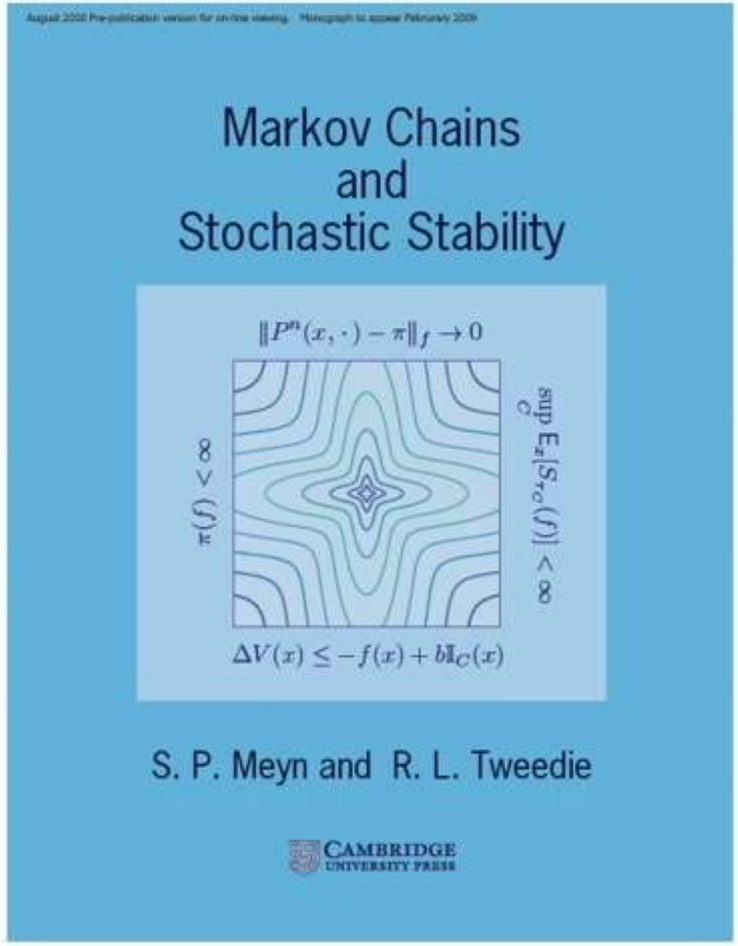
- Part II (Zap Q-learning)

  `https://www.youtube.com/watch?v=Y3w8f1xIb6s`

Other Works

- A. M. Devraj and S. P. Meyn., *Differential TD learning for value function approximation. IEEE Conference on Decision and Control (CDC).* Dec. 2016.
- A. M. Devraj, I. Kontoyiannis and S. P. Meyn., *Geometric ergodicity in a weighted Sobolev space.* Under review, *The Annals of Probability.* Nov. 2017.
- A. M. Devraj, I. Kontoyiannis and S. P. Meyn., *Least squares Differential TD learning.* In preparation. Sep. 2018.


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 February 2009
Markov Chains
and
Stochastic Stability
$\|P^n(x,\cdot) - \pi\|_f \to 0$
$\pi(f) < \infty$
$\sup_C \mathsf{E}_x[S_{\tau_C}(f)] < \infty$
$\Delta V(x) \le -f(x) + b\mathbb{I}_C(x)$
S. P. Meyn and R. L. Tweedie
CAMBRIDGE
UNIVERSITY PRESS

References


Thank you!