Theory of Reinforcement Learning
Aug. 19 – Dec. 18, 2020

Zap Stochastic Approximation

and implications to Q-learning

Sean Meyn

COGNITION & CONTROL IN COMPLEX SYSTEMS

Department of Electrical and Computer Engineering University of Florida

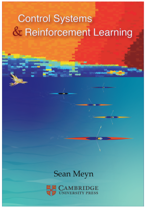
Inria International Chair Inria, Paris

Thanks to to our sponsors: NSF and ARO

Partners in Crime

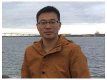
Today's Lecture:

Zap Q Learning with nonlinear function approximation.
S. Chen, A. M. Devraj, A. Bušić, and S. Meyn
NeurIPS, 2020 and arXiv




Shuhang Chen


Adithya Devraj


Fan Lu


Ana Bušić


Crime

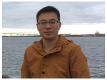
Partners in Crime

Today's Lecture:

Zap Q Learning with nonlinear function approximation.
S. Chen, A. M. Devraj, A. Bušić, and S. Meyn
NeurIPS, 2020   and arXiv

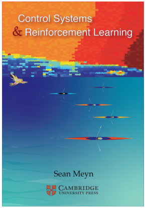

Shuhang Chen


Adithya Devraj


Fan Lu

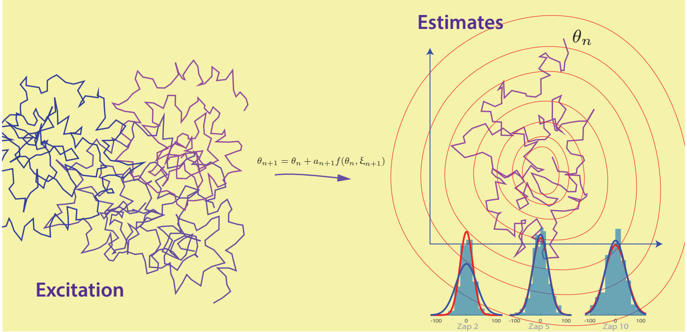

Ana Bušić


Crime



Many Thanks to the Simons Institute for support and inspiration

And thanks to NSF and ARO for supporting this and prior research



Stochastic Approximation

What is Stochastic Approximation? $\bar{f}(\theta) = \mathsf{E}[f(\theta, W)]$

A simple goal: find solution to $\bar{f}(\theta^*) = 0$

What is Stochastic Approximation? $\bar{f}(\theta) = \mathsf{E}[f(\theta, W)]$

A simple goal: find solution to $\bar{f}(\theta^*) = 0$

ODE algorithm: $$\frac{d}{dt}\vartheta_t = \bar{f}(\vartheta_t)$$

If stable: $\vartheta_t \to \theta^*$ and $\bar{f}(\vartheta_t) \to \bar{f}(\theta^*) = 0$.

What is Stochastic Approximation? $\bar{f}(\theta) = \mathsf{E}[f(\theta, W)]$

A simple goal: find solution to $\bar{f}(\theta^*) = 0$

ODE algorithm: $$\frac{d}{dt}\vartheta_t = \bar{f}(\vartheta_t)$$

If stable: $\vartheta_t \to \theta^*$ and $\bar{f}(\vartheta_t) \to \bar{f}(\theta^*) = 0$.

Euler approximation: $$\theta_{n+1} = \theta_n + \alpha_{n+1}\bar{f}(\theta_n)$$

What is Stochastic Approximation? $\bar{f}(\theta) = \mathsf{E}[f(\theta, W)]$

A simple goal: find solution to $\bar{f}(\theta^*) = 0$

ODE algorithm: $$\frac{d}{dt}\vartheta_t = \bar{f}(\vartheta_t)$$

If stable: $\vartheta_t \to \theta^*$ and $\bar{f}(\vartheta_t) \to \bar{f}(\theta^*) = 0$.

Euler approximation: $$\theta_{n+1} = \theta_n + \alpha_{n+1}\bar{f}(\theta_n)$$

Stochastic Approximation

$$\theta_{n+1} = \theta_n + \alpha_{n+1} f(\theta_n, W_{n+1})$$

What is Stochastic Approximation? $\bar{f}(\theta) = \mathsf{E}[f(\theta, W)]$

A simple goal: find solution to $\bar{f}(\theta^*) = 0$

ODE algorithm: $$\frac{d}{dt}\vartheta_t = \bar{f}(\vartheta_t)$$

If stable: $\vartheta_t \to \theta^*$ and $\bar{f}(\vartheta_t) \to \bar{f}(\theta^*) = 0$.

Euler approximation: $$\theta_{n+1} = \theta_n + \alpha_{n+1}\bar{f}(\theta_n)$$

Stochastic Approximation

$$\begin{aligned}\theta_{n+1} &= \theta_n + \alpha_{n+1} f(\theta_n, W_{n+1}) \\ &= \theta_n + \alpha_{n+1}\big\{\bar{f}(\theta_n) + \text{“NOISE”}\big\}\end{aligned}$$

Under very general conditions:

the ODE, the Euler approximation, and SA are all convergent to θ^*

What is Stochastic Approximation? $\bar{f}(\theta) = \mathsf{E}[f(\theta, W)]$

A simple goal: find solution to $\bar{f}(\theta^*) = 0$

ODE algorithm: $$\frac{d}{dt}\vartheta_t = \bar{f}(\vartheta_t)$$

If stable: $\vartheta_t \to \theta^*$ and $\bar{f}(\vartheta_t) \to \bar{f}(\theta^*) = 0$.

Euler approximation: $$\theta_{n+1} = \theta_n + \alpha_{n+1}\bar{f}(\theta_n)$$

Stochastic Approximation

$$\begin{aligned}\theta_{n+1} &= \theta_n + \alpha_{n+1} f(\theta_n, W_{n+1}) \\ &= \theta_n + \alpha_{n+1}\big\{\bar{f}(\theta_n) + \text{“NOISE”}\big\}\end{aligned}$$

Under very general conditions:

the ODE, the Euler approximation, and SA are all convergent to θ^*

Euler approximation is robust to measurement error

What is Stochastic Approximation? $\bar{f}(\theta) = \mathsf{E}[f(\theta, W)]$

A simple goal: find solution to $\bar{f}(\theta^*) = 0$

ODE algorithm: $$\frac{d}{dt}\vartheta_t = \bar{f}(\vartheta_t)$$

If stable: $\vartheta_t \to \theta^*$ and $\bar{f}(\vartheta_t) \to \bar{f}(\theta^*) = 0$.

Euler approximation: $$\theta_{n+1} = \theta_n + \alpha_{n+1}\bar{f}(\theta_n)$$

Stochastic Approximation

$$\begin{aligned} \theta_{n+1} &= \theta_n + \alpha_{n+1} f(\theta_n, W_{n+1}) \\ &= \theta_n + \alpha_{n+1}\big\{\bar{f}(\theta_n) + \text{“NOISE”}\big\} \end{aligned}$$

Under very general conditions:

the ODE, the Euler approximation, and SA are all convergent to θ^*

[Robbins and Monro, 1951] *see Borkar's monograph [5]*

Algorithm Design $\bar{f}(\theta) = \mathsf{E}[f(\theta, W)]$

Stochastic Approximation

$$\begin{aligned}\theta_{n+1} &= \theta_n + \alpha_{n+1} f(\theta_n, W_{n+1}) \\ &= \theta_n + \alpha_{n+1}\big\{\bar{f}(\theta_n) + \text{“NOISE”}\big\}\end{aligned}$$

Step 1: *Design* $\frac{d}{dt}\vartheta_t = \bar{f}(\vartheta_t)$ so that $\vartheta_t \to \theta^*$ and $\bar{f}(\vartheta_t) \to \bar{f}(\theta^*) = 0$.

Algorithm Design $\quad \bar{f}(\theta) = \mathsf{E}[f(\theta, W)]$

Stochastic Approximation

$$\begin{aligned} \theta_{n+1} &= \theta_n + \alpha_{n+1} f(\theta_n, W_{n+1}) \\ &= \theta_n + \alpha_{n+1}\big\{\bar{f}(\theta_n) + \text{“NOISE”}\big\} \end{aligned}$$

Step 1: *Design* $\frac{d}{dt}\vartheta_t = \bar{f}(\vartheta_t)$ so that $\vartheta_t \to \theta^*$ and $\bar{f}(\vartheta_t) \to \bar{f}(\theta^*) = 0$.

You may have to modify the dynamics (spoiler alert!)

Algorithm Design $\bar{f}(\theta) = \mathsf{E}[f(\theta, W)]$

Stochastic Approximation

$$\begin{aligned}\theta_{n+1} &= \theta_n + \alpha_{n+1} f(\theta_n, W_{n+1}) \\ &= \theta_n + \alpha_{n+1}\big\{\bar{f}(\theta_n) + \text{“NOISE”}\big\}\end{aligned}$$

Step 1: *Design* $\frac{d}{dt}\vartheta_t = \bar{f}(\vartheta_t)$ so that $\vartheta_t \to \theta^*$ and $\bar{f}(\vartheta_t) \to \bar{f}(\theta^*) = 0$.
Step 2: Gain selection:
$\alpha_{n+1} = g/(n+1)$ gives optimal convergence rate

$$\mathsf{E}[\|\theta_n - \theta^*\|^2] \approx \frac{1}{n}\mathrm{trace}\,(\Sigma_\theta)$$

Only if $\frac{1}{2}I + gA^*$ is Hurwitz, with $A^* = \partial\bar{f}\,(\theta^*)$

Algorithm Design $\bar{f}(\theta) = \mathsf{E}[f(\theta, W)]$

Stochastic Approximation

$$\begin{aligned}\theta_{n+1} &= \theta_n + \alpha_{n+1} f(\theta_n, W_{n+1}) \\ &= \theta_n + \alpha_{n+1} \big\{ \bar{f}(\theta_n) + \text{"NOISE"} \big\}\end{aligned}$$

Step 1: *Design* $\frac{d}{dt}\vartheta_t = \bar{f}(\vartheta_t)$ so that $\vartheta_t \to \theta^*$ and $\bar{f}(\vartheta_t) \to \bar{f}(\theta^*) = 0$.
Step 2: Gain selection:
$\alpha_{n+1} = g/(n+1)$ gives optimal convergence rate

$$\mathsf{E}[\|\theta_n - \theta^*\|^2] \approx \frac{1}{n} \mathrm{trace}\,(\Sigma_\theta)$$

Only if $\frac{1}{2}I + gA^*$ is Hurwitz, with $A^* = \partial \bar{f}\,(\theta^*)$:

$$0 = [\tfrac{1}{2}I + gA^*]\Sigma_\theta + \Sigma_\theta[\tfrac{1}{2}I + gA^*]^{\intercal} + g^2 \Sigma_{\text{"NOISE"}}$$

$\Longrightarrow$ CLT, etc

SA Error $\theta_{n+1} = \theta_n + \alpha_{n+1}\{\bar{f}(\theta_n) + \text{“NOISE”}\}$ $\frac{d}{dt}\vartheta_t = \bar{f}(\vartheta_t)$

1 $\theta_n - \vartheta_{\tau_n} \approx N(0, \Sigma)$ where

SA Error $\theta_{n+1} = \theta_n + \alpha_{n+1}\{\bar{f}(\theta_n) + \text{"NOISE"}\}$ $\frac{d}{dt}\vartheta_t = \bar{f}(\vartheta_t)$

1 $\theta_n - \vartheta_{\tau_n} \approx N(0,\Sigma)$ where $\tau_n = \sum_{k=1}^{n} \alpha_k$

SA Error $\theta_{n+1} = \theta_n + \alpha_{n+1}\{\bar f(\theta_n) + \text{“NOISE”}\}$ $\frac{d}{dt}\vartheta_t = \bar f(\vartheta_t)$

1 $\theta_n - \vartheta_{\tau_n} \approx N(0,\Sigma)$ where $\tau_n = \sum_{k=1}^{n} \alpha_k$

2 $\vartheta_t \to \theta^*$ exponentially fast, but

SA Error $\theta_{n+1} = \theta_n + \alpha_{n+1}\{\bar{f}(\theta_n) + \text{"NOISE"}\}$ $\frac{d}{dt}\vartheta_t = \bar{f}(\vartheta_t)$

1 $\theta_n - \vartheta_{\tau_n} \approx N(0, \Sigma)$ where $\tau_n = \sum_{k=1}^{n} \alpha_k$

2 $\vartheta_t \to \theta^*$ exponentially fast, but τ_n is increasing slowly, *and*

SA Error $\theta_{n+1} = \theta_n + \alpha_{n+1}\{\bar{f}(\theta_n) + \text{“NOISE”}\}$ $\frac{d}{dt}\vartheta_t = \bar{f}(\vartheta_t)$

1 $\theta_n - \vartheta_{\tau_n} \approx N(0, \Sigma)$ where $\tau_n = \sum_{k=1}^{n} \alpha_k$

2 $\vartheta_t \to \theta^*$ exponentially fast, but τ_n is increasing slowly,
and nonlinear dynamics complicates gain selection

SA Error $\theta_{n+1} = \theta_n + \alpha_{n+1}\{\bar{f}(\theta_n) + \text{"NOISE"}\}$ $\frac{d}{dt}\vartheta_t = \bar{f}(\vartheta_t)$

1 $\theta_n - \vartheta_{\tau_n} \approx N(0,\Sigma)$ where $\tau_n = \sum_{k=1}^{n} \alpha_k$

2 $\vartheta_t \to \theta^*$ exponentially fast, but τ_n is increasing slowly,
and nonlinear dynamics complicates gain selection

What can happen in RL, using $\alpha_{n+1} = g/(n+1)^\rho$:

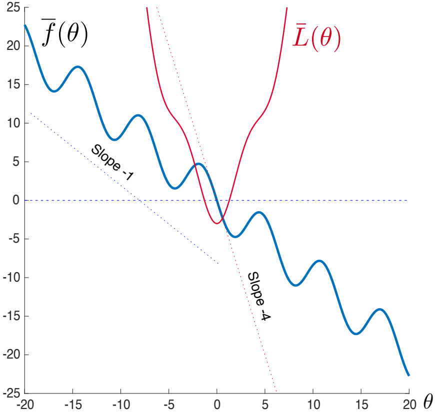
- θ_n far from θ^*, the dynamics are slow, need large g!
- $\theta_n \approx \theta^*$, best gain is far smaller

Two Sources of Error. Example: SGD

Stochastic Gradient Descent:

$$\textcolor{red}{\bar{L}(\theta)} = \mathsf{E}[L(\theta, \Phi_n)]$$

$$\bar{f}(\theta) = -\nabla \bar{L}(\theta)$$



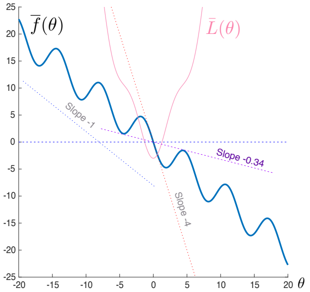
$$\theta_{n+1} = \theta_n - \frac{g}{n^\rho} \nabla L(\theta_n, \Phi_{n+1})$$

Two Sources of Error. Example: SGD

ODE bound using $\rho = 1$

$$|\vartheta_{\tau_n} - \theta^*| \le |\theta_n - \theta^*| e^{0.34 g} n^{-0.34 g}$$

$$\bar f(\theta) = -\nabla \bar L(\theta)$$



$$\theta_{n+1} = \theta_n - \frac{g}{n^\rho} \nabla L(\theta_n, \Phi_{n+1})$$

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

$$|\vartheta_t - \theta^*| \le |\vartheta_0 - \theta^*| e^{-0.34 t}$$

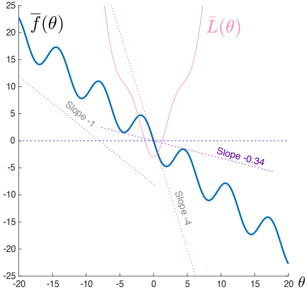
Two Sources of Error. Example: SGD

ODE bound using $\rho = 1$

$$|\vartheta_{\tau_n} - \theta^*| \le |\theta_n - \theta^*| e^{0.34g} n^{-0.34g}$$

$g > 2$ to kill deterministic behavior,
but $g^* = 1/4$ is best locally

$$\bar{f}(\theta) = -\nabla \bar{L}(\theta)$$



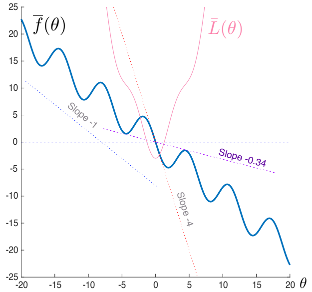
$$\theta_{n+1} = \theta_n - \frac{g}{n^\rho} \nabla L(\theta_n, \Phi_{n+1})$$

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

$$|\vartheta_t - \theta^*| \le |\vartheta_0 - \theta^*| e^{-0.34t}$$

Two Sources of Error. Example: SGD

$$\bar{f}(\theta) = -\nabla \bar{L}(\theta)$$



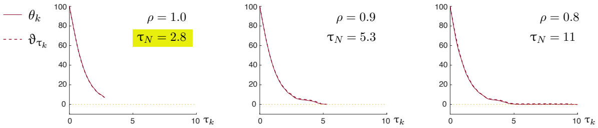
ODE bound using $\rho = 1$

$$|\vartheta_{\tau_n} - \theta^*| \le |\theta_n - \theta^*| e^{0.34g} n^{-0.34g}$$

$g > 2$ to kill deterministic behavior,
but $g^* = 1/4$ is best locally

$$\theta_{n+1} = \theta_n - \frac{g}{n^\rho} \nabla L(\theta_n, \Phi_{n+1})$$

Dynamics for $g^* = 1/4$



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

$$|\vartheta_t - \theta^*| \le |\vartheta_0 - \theta^*| e^{-0.34t}$$

$\tau_N < 3$ for $N =$ *one million*

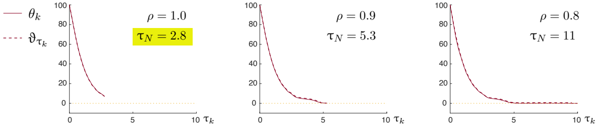
Two Sources of Error. Example: SGD

ODE bound using $\rho = 1$

$$|\vartheta_{\tau_n} - \theta^*| \le |\theta_n - \theta^*| e^{0.34g} n^{-0.34g}$$

$g > 2$ to kill deterministic behavior,
but $g^* = 1/4$ is best locally

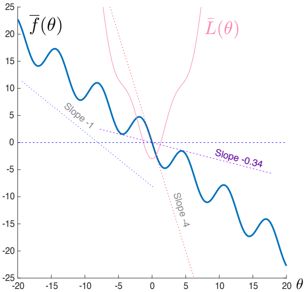
Dynamics for $g^* = 1/4$



$\tau_N < 3$ for $N =$ *one million*

CLT approximation: rapid for $\theta_0 = 0$

$$\bar{f}(\theta) = -\nabla \bar{L}(\theta)$$



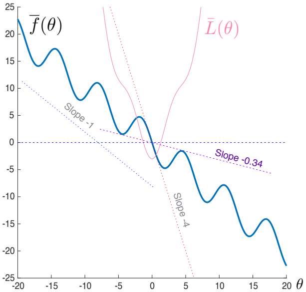
$$\theta_{n+1} = \theta_n - \frac{g}{n^\rho} \nabla L(\theta_n, \Phi_{n+1})$$

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

$$|\vartheta_t - \theta^*| \le |\vartheta_0 - \theta^*| e^{-0.34t}$$

Two Sources of Error. Example: SGD

$$\bar{f}(\theta) = -\nabla \bar{L}(\theta)$$



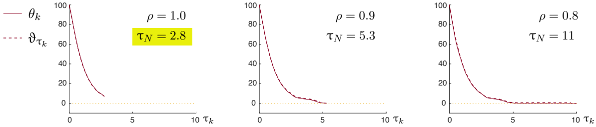
ODE bound using $\rho = 1$

$$|\vartheta_{\tau_n} - \theta^*| \le |\theta_n - \theta^*| e^{0.34g} n^{-0.34g}$$

$g > 2$ to kill deterministic behavior,
but $g^* = 1/4$ is best locally

$$\theta_{n+1} = \theta_n - \frac{g}{n^\rho} \nabla L(\theta_n, \Phi_{n+1})$$

Dynamics for $g^* = 1/4$



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

$$|\vartheta_t - \theta^*| \le |\vartheta_0 - \theta^*| e^{-0.34t}$$

$\tau_N < 3$ for $N =$ *one million*

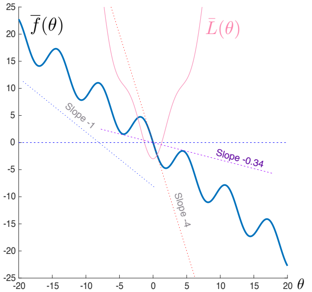
CLT approximation: rapid for $\theta_0 = 0$ slow for $\theta_0 = 100$

Two Sources of Error. Example: SGD

$$\bar{f}(\theta) = -\nabla \bar{L}(\theta)$$

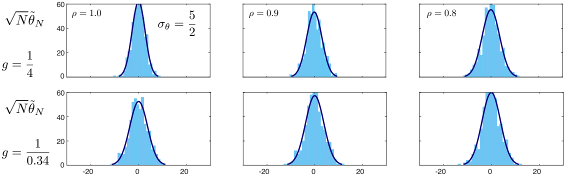
ODE bound using $\rho = 1$

$$|\vartheta_{\tau_n} - \theta^*| \le |\theta_n - \theta^*| e^{0.34g} n^{-0.34g}$$



$$\theta_{n+1} = \theta_n - \frac{g}{n^\rho} \nabla L(\theta_n, \Phi_{n+1})$$

Ruppert-Polyak to the rescue



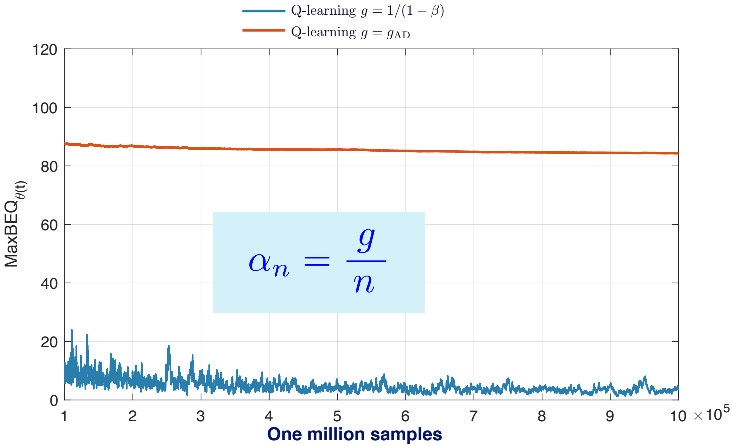
Histograms from Ruppert-Polyak averaging: big and small g

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

$$|\vartheta_t - \theta^*| \le |\vartheta_0 - \theta^*| e^{-0.34t}$$

Two Sources of Error. Example: Tabular Q-Learning

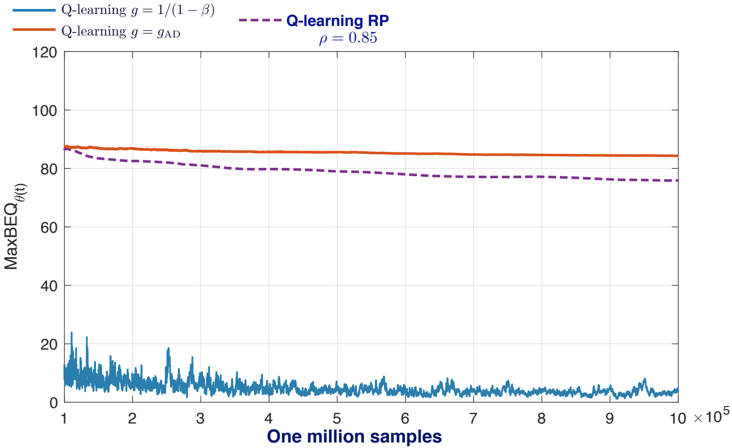
$g \geq 1/(1-\beta)$ required



Generic tabular Q-learning example. Discount factor β

Two Sources of Error. Example: Tabular Q-Learning

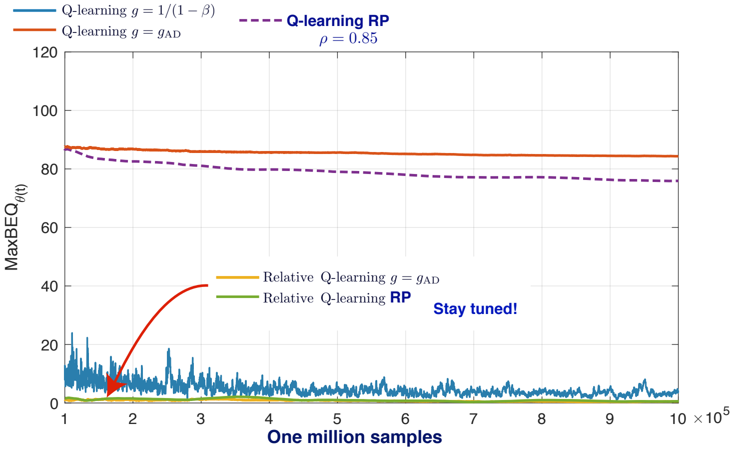
$g \geq 1/(1-\beta)$ required   Ruppert-Polyak to the rescue?



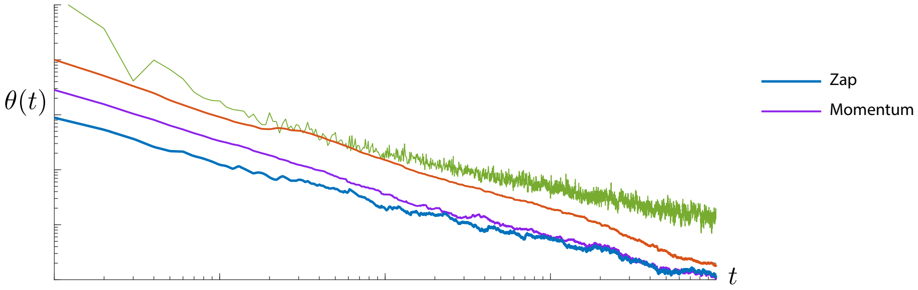
Generic tabular Q-learning example.   Discount factor β

Two Sources of Error. Example: Tabular Q-Learning

Ruppert-Polyak to the rescue? Culprit is Condition Number



Generic tabular Q-learning example. Discount factor β



Return to Zap

Motivation

ODE Design begins with design of the ODE: $\frac{d}{dt}\vartheta = \bar{f}(\vartheta)$
Challenges we have faced with Q-learning:

Motivation

ODE Design begins with design of the ODE: $\frac{d}{dt}\vartheta = \bar{f}(\vartheta)$
Challenges we have faced with Q-learning:

- How can we design dynamics for
  1. Stability
  2. $\bar{f}(\theta^*) = 0$ solves a relevant problem

Motivation

ODE Design begins with design of the ODE: $\frac{d}{dt}\vartheta = \bar{f}(\vartheta)$
Challenges we have faced with Q-learning:

- How can we design dynamics for
  1. Stability
  2. $\bar{f}(\theta^*) = 0$ solves a relevant problem
- How can we better manage problems introduced by $1/(1-\beta)$?

Motivation

ODE Design begins with design of the ODE: $\frac{d}{dt}\vartheta = \bar{f}(\vartheta)$
Challenges we have faced with Q-learning:

- How can we design dynamics for
  1. Stability
  2. $\bar{f}(\theta^*) = 0$ solves a relevant problem
- How can we better manage problems introduced by $1/(1-\beta)$?

Assuming we have solved 2, maybe we can create linear dynamics (Newton-Raphson flow):

$$\frac{d}{dt}\bar{f}(\vartheta_t) = -\bar{f}(\vartheta_t) \qquad \textit{giving} \quad \bar{f}(\vartheta_t) = \bar{f}(\vartheta_0)e^{-t}$$

Smale 1976

Motivation

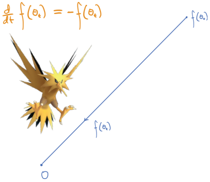
ODE Design begins with design of the ODE: $\frac{d}{dt}\vartheta = \bar{f}(\vartheta)$

Challenges we have faced with Q-learning:

- How can we design dynamics for
  1. Stability
  2. $\bar{f}(\theta^*) = 0$ solves a relevant problem
- How can we better manage problems introduced by $1/(1-\beta)$?

Assuming we have solved 2, maybe we can create linear dynamics (Newton-Raphson flow):

$$\frac{d}{dt}\bar{f}(\vartheta_t) = -\bar{f}(\vartheta_t) \qquad \textit{giving} \quad \bar{f}(\vartheta_t) = \bar{f}(\vartheta_0)e^{-t}$$



The SA translation is Zap Stochastic Approximation

Zap Algorithm

Newton-Raphson flow: $\frac{d}{dt}\vartheta_t = -A(\vartheta_t)^{-1}\bar{f}(\vartheta_t), \quad A(\theta) = \frac{\partial}{\partial\theta}\bar{f}(\theta)$

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

$$\theta_{n+1} = \theta_n + \alpha_{n+1}(-\widehat{A}_{n+1})^{-1} f(\theta_n, \Phi_{n+1})$$

$$\widehat{A}_{n+1} = \widehat{A}_n + \gamma_{n+1}(A_{n+1} - \widehat{A}_n), \qquad A_{n+1} = \partial_\theta f(\theta_n, \Phi_{n+1})$$

Zap Algorithm

Newton-Raphson flow: $\frac{d}{dt}\vartheta_t = -A(\vartheta_t)^{-1}\bar{f}(\vartheta_t), \quad A(\theta) = \frac{\partial}{\partial\theta}\bar{f}(\theta)$

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

$$\theta_{n+1} = \theta_n + \alpha_{n+1}(-\widehat{A}_{n+1})^{-1} f(\theta_n, \Phi_{n+1})$$

$$\widehat{A}_{n+1} = \widehat{A}_n + \gamma_{n+1}(A_{n+1} - \widehat{A}_n), \qquad A_{n+1} = \partial_\theta f(\theta_n, \Phi_{n+1})$$

Requires $\widehat{A}_{n+1} \approx A(\theta_n) \stackrel{\text{def}}{=} \partial_\theta \bar{f}\,(\theta_n)$

Zap Algorithm

Newton-Raphson flow: $\frac{d}{dt}\vartheta_t = -A(\vartheta_t)^{-1}\bar{f}(\vartheta_t), \quad A(\theta) = \frac{\partial}{\partial\theta}\bar{f}(\theta)$

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

$$\theta_{n+1} = \theta_n + \alpha_{n+1}(-\widehat{A}_{n+1})^{-1} f(\theta_n, \Phi_{n+1})$$

$$\widehat{A}_{n+1} = \widehat{A}_n + \gamma_{n+1}(A_{n+1} - \widehat{A}_n), \qquad A_{n+1} = \partial_\theta f(\theta_n, \Phi_{n+1})$$

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

Zap Algorithm

Newton-Raphson flow: $\frac{d}{dt}\vartheta_t = -A(\vartheta_t)^{-1}\bar{f}(\vartheta_t), \quad A(\theta) = \frac{\partial}{\partial\theta}\bar{f}(\theta)$

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

$$\theta_{n+1} = \theta_n + \alpha_{n+1}(-\widehat{A}_{n+1})^{-1} f(\theta_n, \Phi_{n+1})$$
$$\widehat{A}_{n+1} = \widehat{A}_n + \gamma_{n+1}(A_{n+1} - \widehat{A}_n), \qquad A_{n+1} = \partial_\theta f(\theta_n, \Phi_{n+1})$$

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

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

Zap Algorithm

Newton-Raphson flow: $\frac{d}{dt}\vartheta_t = -A(\vartheta_t)^{-1}\bar{f}(\vartheta_t), \quad A(\theta) = \frac{\partial}{\partial\theta}\bar{f}(\theta)$

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

$$\theta_{n+1} = \theta_n + \alpha_{n+1}(-\widehat{A}_{n+1})^{-1} f(\theta_n, \Phi_{n+1})$$

$$\widehat{A}_{n+1} = \widehat{A}_n + \gamma_{n+1}(A_{n+1} - \widehat{A}_n), \qquad A_{n+1} = \partial_\theta f(\theta_n, \Phi_{n+1})$$

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

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

Stability? *Virtually universal*

Optimal variance, too!
Based on ancient theory from Ruppert & Polyak [10, 11, 9]

Zap Q-Learning

Q-learning: $\{Q^{\theta}(x,u) : \theta \in \mathbb{R}^d\,,\ u \in \mathsf{U}\,,\ x \in \mathsf{X}\}$

Find θ^* such that $\bar{f}(\theta^*) = 0$, with

Zap Q-Learning

Q-learning: $\{Q^\theta(x,u) : \theta \in \mathbb{R}^d\,,\; u \in \mathsf{U}\,,\; x \in \mathsf{X}\}$

Find θ^* such that $\bar{f}(\theta^*) = 0$, with

$$\bar{f}(\theta) = \mathsf{E}\Big[\big\{c(X_n,U_n) + \beta \underline{Q}^\theta(X_{n+1}) - Q^\theta(X_n,U_n)\big\}\zeta_n\Big]$$

Example: $\zeta_n = \nabla_\theta Q^\theta(X_n,U_n)$

Zap Q-Learning

Q-learning: $\{Q^\theta(x,u) : \theta \in \mathbb{R}^d\,,\ u \in \mathsf{U}\,,\ x \in \mathsf{X}\}$

Find θ^* such that $\bar{f}(\theta^*) = 0$, with

$$\bar{f}(\theta) = \mathsf{E}\Big[\Big\{c(X_n,U_n) + \beta \underline{Q}^\theta(X_{n+1}) - Q^\theta(X_n,U_n)\Big\}\zeta_n\Big]$$

Example: $\zeta_n = \nabla_\theta Q^\theta(X_n,U_n)$

This is the hidden goal of DQN

Zap Q-Learning

Q-learning: $\{Q^\theta(x,u) : \theta \in \mathbb{R}^d\,,\ u \in \mathsf{U}\,,\ x \in \mathsf{X}\}$

Find θ^* such that $\bar{f}(\theta^*) = 0$, with

$$\bar{f}(\theta) = \mathsf{E}\big[\big\{c(X_n,U_n) + \beta \underline{Q}^\theta(X_{n+1}) - Q^\theta(X_n,U_n)\big\}\zeta_n\big]$$

Example: $\zeta_n = \nabla_\theta Q^\theta(X_n,U_n)$

This is the hidden goal of DQN

What makes theory difficult:

1. Does $\bar{f}$ have a root?
2. Does the inverse of A exist?
3. SA theory is weak for a discontinuous ODE

Zap Q-Learning

Q-learning: $\{Q^\theta(x,u) : \theta \in \mathbb{R}^d\,,\ u \in \mathsf{U}\,,\ x \in \mathsf{X}\}$

Find θ^* such that $\bar{f}(\theta^*) = 0$, with

$$\bar{f}(\theta) = \mathsf{E}\big[\{c(X_n,U_n) + \beta \underline{Q}^\theta(X_{n+1}) - Q^\theta(X_n,U_n)\}\zeta_n\big]$$

Example: $\zeta_n = \nabla_\theta Q^\theta(X_n,U_n)$

This is the hidden goal of DQN

What makes theory difficult:

1. Does $\bar{f}$ have a root?
2. Does the inverse of A exist?
3. SA theory is weak for a discontinuous ODE

See NeurIPS video for 2 and 3 (and [1])

Zap Examples

$$0 = \bar{f}(\theta^*) = \mathsf{E}\Big[\big\{c(X_n, U_n) + \beta \underline{Q}^{\theta^*}(X_{n+1}) - Q^{\theta^*}(X_n, U_n)\big\}\zeta_n\Big]$$

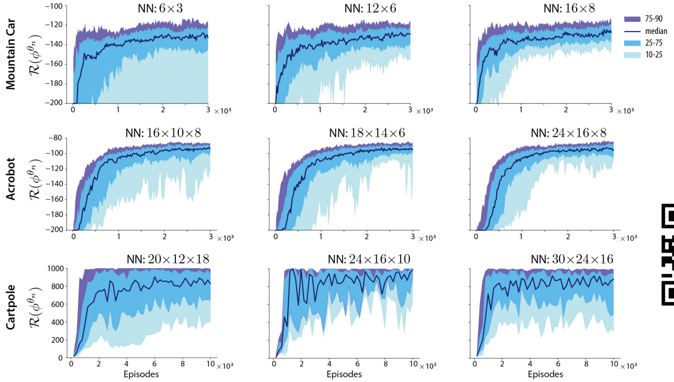
$$\zeta_n = \nabla_\theta Q^\theta(X_n, U_n)\big|_{\theta=\theta^*}$$

Zap Examples

$$0 = \bar{f}(\theta^*) = \mathsf{E}\Big[\big\{c(X_n, U_n) + \beta \underline{Q}^{\theta^*}(X_{n+1}) - Q^{\theta^*}(X_n, U_n)\big\}\zeta_n\Big]$$

$$\zeta_n = \nabla_\theta Q^\theta(X_n, U_n)\big|_{\theta=\theta^*}$$

VI. Stunning reliability with Q^θ parameterized by various neural networks



Conclusions:

- Reinforcement Learning is cursed by dimension, variance, and nonlinear (algorithm) dynamics

Conclusions:

- Reinforcement Learning is cursed by dimension, variance, and nonlinear (algorithm) dynamics
- Second order methods can ensure stability—use them when you can

Conclusions:

- Reinforcement Learning is cursed by dimension, variance, and nonlinear (algorithm) dynamics
- Second order methods can ensure stability—use them when you can

Future work:

- Beyond the projected Bellman error for Q-learning
- Applications in Stochastic Optimization

Conclusions:

- Reinforcement Learning is cursed by dimension, variance, and nonlinear (algorithm) dynamics
- Second order methods can ensure stability—use them when you can

Future work:

- Beyond the projected Bellman error for Q-learning
- Applications in Stochastic Optimization
- Acceleration techniques (momentum and matrix momentum)

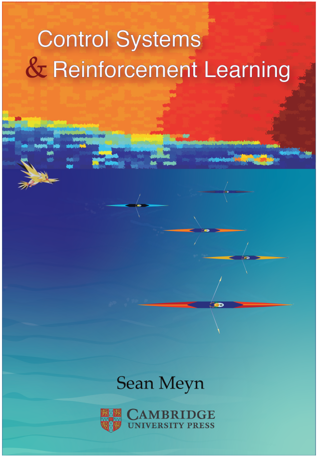
Conclusions:

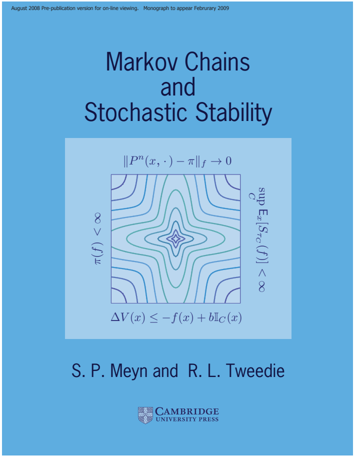
- Reinforcement Learning is cursed by dimension, variance, and nonlinear (algorithm) dynamics
- Second order methods can ensure stability—use them when you can

Future work:

- Beyond the projected Bellman error for Q-learning
- Applications in Stochastic Optimization
- Acceleration techniques (momentum and matrix momentum)
- Further variance reduction using control variates


Thank you!


Control Systems
& Reinforcement Learning
Sean Meyn
CAMBRIDGE
UNIVERSITY PRESS


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

References

Selected References I

[1] S. Chen, A. M. Devraj, A. Bušić, and S. Meyn. Zap Q Learning with nonlinear function approximation. To appear NeurIPS and arXiv e-prints 1910.05405, 2020.

[2] A. M. Devraj, A. Bušić, and S. Meyn. Fundamental design principles for reinforcement learning algorithms. In *Handbook on Reinforcement Learning and Control*. Springer, 2020.

[3] A. M. Devraj. *Reinforcement Learning Design with Optimal Learning Rate*. PhD thesis, University of Florida, 2019.

[4] S. Smale. A convergent process of price adjustment and global Newton methods. *Journal of Mathematical Economics*, 3(2):107–120, July 1976.

Stochastic Approximation

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

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

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

Selected References II

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

[9] 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.

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

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

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

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

Selected References III

[14] F. R. Bach and E. Moulines Non-strongly-convex smooth stochastic approximation with convergence rate $O(1/n)$. In *Advances in Neural Information Processing Systems 24*, 773–781. Curran Associates, Inc., 2013.

[15] S. Chen, A. M. Devraj, A. Bušić, and S. Meyn. Explicit mean-square error bounds for Monte-Carlo and linear stochastic approximation. *AISTATS*, page arXiv:2002.02584, Feb. 2020.

[16] A. M. Devraj, A. Bušić, and S. Meyn. On matrix momentum stochastic approximation and applications to Q-learning. In *Allerton Conference on Communication, Control, and Computing*, pages 749–756, Sep 2019.