Why Does Q-learning Work?

The Projected Bellman Equation in Reinforcement 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

Why Does Q-learning Work?

Outline

Background: Stochastic Approximation

ODE Method (using different meaning than in the 1970s)

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

Background: Stochastic Approximation

ODE Method (using different meaning than in the 1970s)

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

$$\bar{f}(\theta) = \mathsf{E}[f(\theta, \xi_{n+1})]$$

Background: Stochastic Approximation



ODE Method (using different meaning than in the 1970s)

Goal: *find solution to* $\bar{f}(\theta^*) = 0$ $\qquad \bar{f}(\theta) = \mathsf{E}[f(\theta, \xi_{n+1})]$

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

design for stability

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

Background: Stochastic Approximation



ODE Method

(using different meaning than in the 1970s)

Goal: *find solution to* $\bar{f}(\theta^*) = 0$ $\qquad\qquad$ $\bar{f}(\theta) = \mathsf{E}[f(\theta, \xi_{n+1})]$

ODE algorithm: $\displaystyle \frac{d}{dt}\vartheta_t = \bar{f}(\vartheta_t)$ $\qquad$ design for stability

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

Stochastic Approximation: $\theta_{n+1} = \theta_n + \alpha_{n+1} f(\theta_n, \xi_{n+1})$

Background: Stochastic Approximation



ODE Method (using different meaning than in the 1970s)

Goal: *find solution to* $\bar{f}(\theta^*) = 0$ $\bar{f}(\theta) = \mathsf{E}[f(\theta, \xi_{n+1})]$

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

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

Stochastic Approximation: $\theta_{n+1} = \theta_n + \alpha_{n+1} f(\theta_n, \xi_{n+1})$

$f(\theta_n, \xi_{n+1})$: the *reinforcement signal* in Sutton's early work [18]

Background: Stochastic Approximation



ODE Method (using different meaning than in the 1970s)

Goal: *find solution to* $\bar{f}(\theta^*) = 0$ $\quad\quad$ $\bar{f}(\theta) = \mathsf{E}[f(\theta, \xi_{n+1})]$

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

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

Stochastic Approximation: $\theta_{n+1} = \theta_n + \alpha_{n+1} f(\theta_n, \xi_{n+1})$

$f(\theta_n, \xi_{n+1})$: the *reinforcement signal* in Sutton's early work [18]

Firm theory of RL based on SA initiated in Tsitsiklis, 1994 [21]

Resources

ODE Method (using different meaning than in the 1970s)

- CS&RL, Chapters 4 and 8
- The ODE Method for Asymptotic Statistics in Stochastic Approximation and Reinforcement Learning [90, 92]
  And of course *Borkar's manifesto* [64]

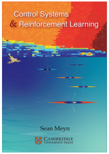
Resources

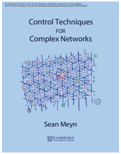
ODE Method (using different meaning than in the 1970s)

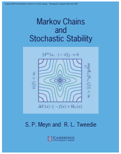
- CS&RL, Chapters 4 and 8
- The ODE Method for Asymptotic Statistics in Stochastic Approximation and Reinforcement Learning [90, 92]
  And of course *Borkar's manifesto* [64]

TD Methods CS&RL:

- Chapter 5 (purely deterministic setting)
- Chapters 9 & 10 (traditional MDP)







Resources

ODE Method (using different meaning than in the 1970s)

- CS&RL, Chapters 4 and 8
- The ODE Method for Asymptotic Statistics in Stochastic Approximation and Reinforcement Learning [90, 92]
  And of course *Borkar's manifesto* [64]

TD Methods CS&RL:

- Chapter 5 (purely deterministic setting)
- Chapters 9 & 10 (traditional MDP)

New material in this lecture:

[9] *The projected Bellman equation in reinforcement learning. IEEE Transactions on Automatic Control*, 2024.
[10] *Stability of Q-learning through design and optimism. arXiv 2307.02632*, 2023.

Resources



Related literature: in addition to tabular [20, 19] and binning [22]

[11] D. De Farias and B. Van Roy. On the existence of fixed points for approximate value iteration and temporal-difference learning, 2000.

[12] Z. Chen, J.-P. Clarke, and S. T. Maguluri. Target network and truncation overcome the deadly triad in Q-learning, 2023.

[14] F. S. Melo, S. P. Meyn, and M. I. Ribeiro. An analysis of reinforcement learning with function approximation, 2008.

[13] D. Lee and N. He. A unified switching system perspective and convergence analysis of Q-learning algorithms, 2020.

New material in this lecture:

[9] *The projected Bellman equation in reinforcement learning. IEEE Transactions on Automatic Control*, 2024.

[10] *Stability of Q-learning through design and optimism. arXiv 2307.02632*, 2023.

Resources



Related literature: in addition to tabular [20, 19] and binning [22]

[11] D. De Farias and B. Van Roy. On the existence of fixed points for approximate value iteration and temporal-difference learning, 2000. Existence of θ^* for "ε-optimistic SARSA"

[12] Z. Chen, J.-P. Clarke, and S. T. Maguluri. Target network and truncation overcome the deadly triad in Q-learning, 2023. Bounds on iterates for a projected temporal difference

[14] F. S. Melo, S. P. Meyn, and M. I. Ribeiro. An analysis of reinforcement learning with function approximation, 2008. Semi-circular conditions for convergence

[13] D. Lee and N. He. A unified switching system perspective and convergence analysis of Q-learning algorithms, 2020. A fresh take on [14]

New material in this lecture:

[9] *The projected Bellman equation in reinforcement learning. IEEE Transactions on Automatic Control*, 2024.
[10] *Stability of Q-learning through design and optimism. arXiv 2307.02632*, 2023.

Resources



Related literature: See Zap for convergence without linear function approx. [38]

[11] D. De Farias and B. Van Roy. On the existence of fixed points for approximate value iteration and temporal-difference learning, 2000. Existence of θ^* for "ε-optimistic SARSA"

[12] Z. Chen, J.-P. Clarke, and S. T. Maguluri. Target network and truncation overcome the deadly triad in Q-learning, 2023. Bounds on iterates for a projected temporal difference

[14] F. S. Melo, S. P. Meyn, and M. I. Ribeiro. An analysis of reinforcement learning with function approximation, 2008. Semi-circular conditions for convergence

[13] D. Lee and N. He. A unified switching system perspective and convergence analysis of Q-learning algorithms, 2020. A fresh take on [14]

New material in this lecture:

[9] *The projected Bellman equation in reinforcement learning. IEEE Transactions on Automatic Control*, 2024.

[10] *Stability of Q-learning through design and optimism. arXiv 2307.02632*, 2023.

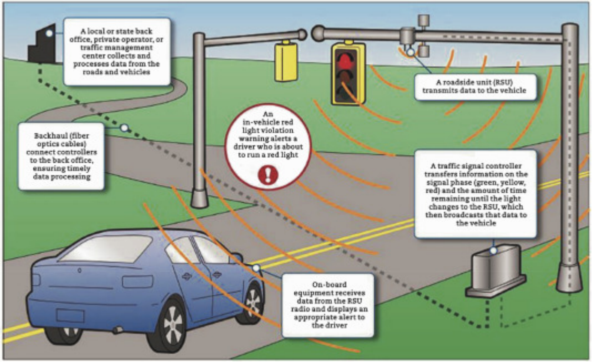
Too many resources to list

Sadly, I am leaving out all of the fun **zero-variance** theory with Caio Lauand

Introducing Dr. Lauand in May, 2025

Very partial list of publications: (left out the two neurips pubs)

- Quasi-stochastic approximation: Design principles with applications to extremum seeking control, 2023, [79]
- The curse of memory in stochastic approximation, 2023, [93]
- Markovian foundations for quasi stochastic approximation, 2024, [80]
- Revisiting step-size assumptions in stochastic approximation, 2024, [92]


A local or state back office, private operator, or traffic management center collects and processes data from the roads and vehicles
A roadside unit (RSU) transmits data to the vehicle
An in-vehicle red light violation warning alerts a driver who is about to run a red light
Backhaul (fiber optics cables) connect controllers to the back office, ensuring timely data processing
A traffic signal controller transfers information on the signal phase (green, yellow, red) and the amount of time remaining until the light changes to the RSU, which then broadcasts that data to the vehicle
On-board equipment receives data from the RSU radio and displays an appropriate alert to the driver

Q Learning

Stochastic Optimal Control (Review)

MDP Model

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

- For all states x and sets A,

$$\mathsf{P}\{X_{n+1} \in A \mid X_n = x,\ U_n = u, \text{and prior history}\} = P_u(x, A)$$

- $c\colon \mathsf{X} \times \mathsf{U} \to \mathbb{R}$ is a cost function
- $\gamma < 1$ a discount factor

Q function:

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

Stochastic Optimal Control (Review)

MDP Model

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

- For all states x and sets A,

$$\mathsf{P}\{X_{n+1} \in A \mid X_n = x,\ U_n = u, \text{and prior history}\} = P_u(x, A)$$

- $c\colon \mathsf{X} \times \mathsf{U} \to \mathbb{R}$ is a cost function
- $\gamma < 1$ a discount factor

Q function:

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

Bellman equation:

$$Q^*(x,u) = c(x,u) + \gamma \mathsf{E}\big[\min_{u'} Q^*(X_1, u') \mid X_0 = x,\ U_0 = u\big]$$

Q-Learning and Galerkin Relaxation

Dynamic programming

Find function Q^* that solves (${\cal F}_n$ means history)

$$\mathsf{E}\Big[c(X_n,U_n)+\gamma\underline{Q}^*(X_{n+1})-Q^*(X_n,U_n)\mid {\cal F}_n\Big]=0$$

$$\underline{H}(x)=\min_u H(x,u)$$

Q-Learning and Galerkin Relaxation

Dynamic programming

Find function Q^* that solves ($\mathcal{F}_n$ means history)

$$\mathsf{E}\big[c(X_n,U_n) + \gamma \underline{Q}^*(X_{n+1}) - Q^*(X_n,U_n) \mid \mathcal{F}_n\big] = 0$$

Goal of Q-Learning

Given $\{Q^\theta : \theta \in \mathbb{R}^d\}$, find θ^* that solves $\bar{f}(\theta^*) = 0$,

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

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

Q-Learning and Galerkin Relaxation

Dynamic programming

Find function Q^* that solves ($\mathcal{F}_n$ means history)

$$\mathsf{E}\big[c(X_n, U_n) + \gamma \underline{Q}^*(X_{n+1}) - Q^*(X_n, U_n) \mid \mathcal{F}_n\big] = 0$$

Goal of Q-Learning

Given $\{Q^\theta : \theta \in \mathbb{R}^d\}$, find θ^* that solves $\bar{f}(\theta^*) = 0$,

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

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

Projected Bellman Equation: $\bar{f}(\theta^*) = 0$

Q(0)-Learning Goal $\bar{f}(\theta^*) = 0$

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

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

Q(0)-Learning

Goal $\bar{f}(\theta^*) = 0$

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

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

$\Longrightarrow$ prototypical Q-learning algorithm

Q(0) Learning Algorithm

Estimates obtained using SA

$$\theta_{n+1} = \theta_n + \alpha_{n+1} f_{n+1} \qquad f_{n+1} = \{c_n + \gamma \underline{Q}^\theta_{n+1} - Q^\theta_n\}\zeta_n\Big|_{\theta=\theta_n}$$

$$\underline{Q}^\theta_{n+1} = Q^\theta(X_{n+1}, \phi^\theta(X_{n+1}))$$

- $\phi^\theta(x) = \arg\min_u Q^\theta(x,u)$ [Q^θ-greedy policy]
- Input $\{U_n\}$ chosen for *exploration*.

Q(0)-Learning Goal $\bar{f}(\theta^*) = 0$

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

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

$\Longrightarrow$ prototypical Q-learning algorithm

Q(0) Learning Algorithm

Estimates obtained using SA

$$\theta_{n+1} = \theta_n + \alpha_{n+1} f_{n+1} \qquad f_{n+1} = \{c_n + \gamma \underline{Q}^\theta_{n+1} - Q^\theta_n\}\zeta_n\Big|_{\theta=\theta_n}$$

$$\underline{Q}^\theta_{n+1} = Q^\theta(X_{n+1}, \phi^\theta(X_{n+1}))$$

- $\phi^\theta(x) = \arg\min_u Q^\theta(x, u)$ [Q^θ-greedy policy]
- Input $\{U_n\}$ chosen for *exploration*.
  Oblivious if independent of θ_n (in which case usually i.i.d.)

Q(0)-Learning Goal $\bar{f}(\theta^*) = 0$

$Q(0)$-learning with linear function approximation

Estimates obtained using SA

$$\theta_{n+1} = \theta_n + \alpha_{n+1} f_{n+1} \qquad f_{n+1} = \Big\{ c_n + \gamma \underline{Q}^\theta_{n+1} - Q^\theta_n \Big\}\Big|_{\theta=\theta_n} \zeta_n$$

$$\underline{Q}^\theta_{n+1} = Q^\theta(X_{n+1}), \phi^\theta(X_{n+1}))$$

- $Q^\theta(x,u) = \theta^\intercal \psi(x,u)$
- $\underline{Q}^\theta(x) = \theta^\intercal \psi(x, \phi^\theta(x))$
- $\zeta_n = \nabla_\theta Q^\theta(X_n, U_n)\big|_{\theta=\theta_n} = \psi(X_n, U_n)$

Q(0)-Learning

Goal $\bar{f}(\theta^*) = 0$

$Q(0)$-learning with linear function approximation

Estimates obtained using SA

$$\theta_{n+1} = \theta_n + \alpha_{n+1} f_{n+1} \qquad f_{n+1} = \{c_n + \gamma \underline{Q}^\theta_{n+1} - Q^\theta_n\}\Big|_{\theta=\theta_n} \zeta_n$$

$$\underline{Q}^\theta_{n+1} = Q^\theta(X_{n+1}), \phi^\theta(X_{n+1}))$$

- $Q^\theta(x,u) = \theta^\intercal \psi(x,u)$
- $\underline{Q}^\theta(x) = \theta^\intercal \psi(x, \phi^\theta(x))$
- $\zeta_n = \nabla_\theta Q^\theta(X_n, U_n)\big|_{\theta=\theta_n} = \psi(X_n, U_n)$

$\bar{f}(\theta) = \overline{A}(\theta)\theta - \bar{b}$

p.w. constant if U is oblivious

$$\overline{A}(\theta) = \mathsf{E}\big[\zeta_n\big[\gamma\psi(X_{n+1}, \phi^\theta(X_{n+1})) - \psi(X_n, U_n)\big]^\intercal\big]$$

$$\bar{b} \stackrel{\text{def}}{=} \mathsf{E}\big[\zeta_n c(X_n, U_n)\big]$$

Watkins' Q-learning

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

Watkins' Q-learning

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

Watkin's algorithm *A special case of Q(0)-learning*

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)

Convergence of Q^{θ_n} to Q^* holds under mild conditions

Watkins' Q-learning

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

Watkin's algorithm *A special case of Q(0)-learning*

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)

Convergence of Q^{θ_n} to Q^* holds under mild conditions

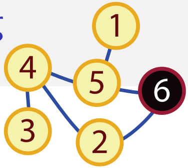
Asymptotic covariance is infinite for $\gamma\geq 1/2$ [37]

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

Using the standard step-size rule $\alpha_n=1/n(x,u)$

Asymptotic Covariance of Watkins' Q-Learning

This is what infinite variance looks like

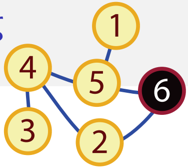


$$\sigma^2 = \lim_{n\to\infty} n\mathsf{E}[\|\theta_n - \theta^*\|^2] = \infty$$

Wild oscillations?

Asymptotic Covariance of Watkins' Q-Learning

This is what infinite variance looks like

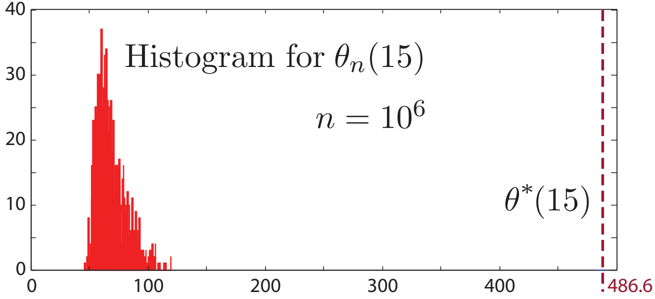


$$\sigma^2 = \lim_{n\to\infty} n\mathsf{E}[\|\theta_n - \theta^*\|^2] = \infty$$

Wild oscillations?

Not at all, the sample paths appear frozen

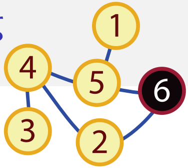
Histogram of parameter estimates after 10^6 iterations.



Example from [37] 2017

Asymptotic Covariance of Watkins' Q-Learning

This is what infinite variance looks like



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

Wild oscillations?

Not at all, the sample paths appear frozen

Sample paths using a higher gain, or relative Q-learning [74, 76]

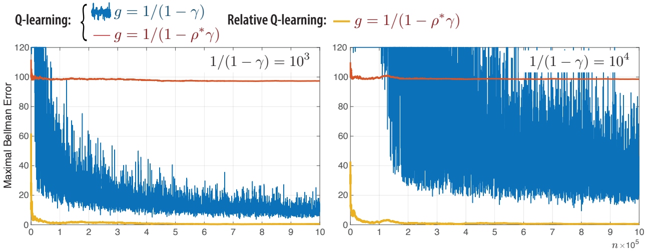
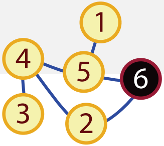


Figure 1: Comparison of Q-learning and Relative Q-learning algorithms for the stochastic shortest path problem of [4]. The relative Q-learning algorithm is unaffected by large discounting.

Example from [37] 2017, and [74, 76], CS&RL, 2021

Asymptotic Covariance of Watkins' Q-Learning

Can we do better?



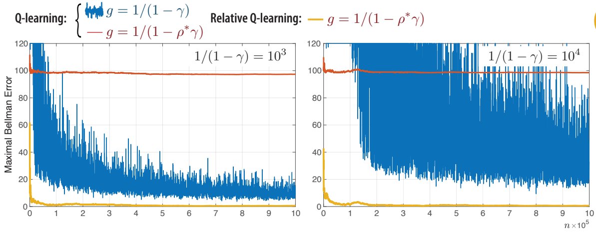


Figure 1: Comparison of Q-learning and Relative Q-learning algorithms for the stochastic shortest path problem of [4]. The relative Q-learning algorithm is unaffected by large discounting.

Relative Q-learning: estimate *relative Q-function*,

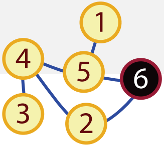
$$\mathsf{E}\big[c(X_n,U_n) + \gamma \underline{H}^*(X_{n+1}) - H^*(X_n,U_n) - \gamma\langle \nu, H^*\rangle \mid \mathcal{F}_n\big] = 0$$

giving $H^* = Q^* + \text{const.}$ [74, 76]

And don't use step-size $\alpha_n = g/n$ (see SA tutorial)

Asymptotic Covariance of Watkins' Q-Learning

Can we do better?



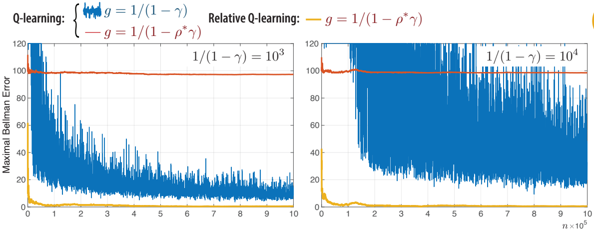
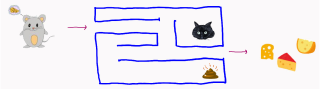


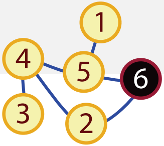
Figure 1: Comparison of Q-learning and Relative Q-learning algorithms for the stochastic shortest path problem of [4]. The relative Q-learning algorithm is unaffected by large discounting.

An intelligent mouse might offer other clues



Asymptotic Covariance of Watkins' Q-Learning

Can we do better?



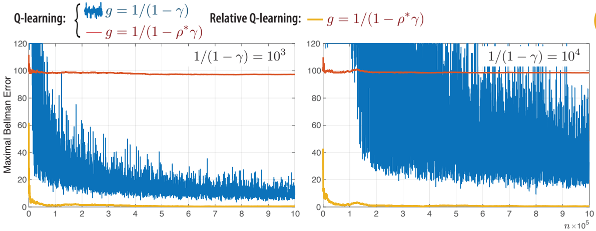
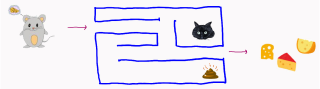


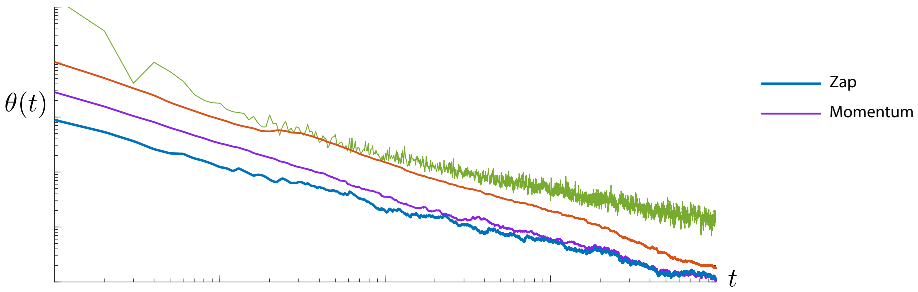
Figure 1: Comparison of Q-learning and Relative Q-learning algorithms for the stochastic shortest path problem of [4]. The relative Q-learning algorithm is unaffected by large discounting.

An intelligent mouse might offer other clues



First consider second order methods

or ⏩ Skip to newest theory



Zap

Motivation

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

Motivation

The ODE method 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 few results outside of Watkins' tabular setting
  2. $\bar{f}(\theta^*) = 0$ solves a relevant problem or has a solution

Motivation

The ODE method 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 $1/(1-\gamma)$?

Relative Q-Learning is one approach

Motivation

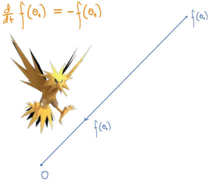
The ODE method begins with design of the ODE: $\frac{d}{dt}\vartheta = \bar{f}(\vartheta)$
Challenges we have faced with Q-learning:

- 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 $1/(1-\gamma)$?

Relative Q-Learning is one approach

Assuming we have solved 2, forget 1 and
approximate Newton-Raphson flow:

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



Zap Algorithm

Designed to emulate Newton-Raphson flow

$$\frac{d}{dt}\vartheta_t = -[A(\vartheta_t)]^{-1}\bar{f}(\vartheta_t), \quad A(\theta) = \partial_\theta \bar{f}\,(\theta)$$

Zap-SA

$$\theta_{n+1} = \theta_n + \alpha_{n+1} G_{n+1} f(\theta_n, \Xi_{n+1}) \qquad G_{n+1} = -[\widehat{A}_{n+1}]^{-1}$$

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

Zap Algorithm

Designed to emulate Newton-Raphson flow

$\frac{d}{dt}\vartheta_t = -[A(\vartheta_t)]^{-1}\bar{f}(\vartheta_t), \quad A(\theta) = \partial_\theta \bar{f}(\theta)$

Zap-SA

$$\theta_{n+1} = \theta_n + \alpha_{n+1} G_{n+1} f(\theta_n, \xi_{n+1}) \qquad G_{n+1} = -[\widehat{A}_{n+1}]^{-1}$$

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

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

Zap Algorithm

Designed to emulate Newton-Raphson flow

$$\frac{d}{dt}\vartheta_t = -[A(\vartheta_t)]^{-1}\bar{f}(\vartheta_t), \quad A(\theta) = \partial_\theta \bar{f}(\theta)$$

Zap-SA

$$\theta_{n+1} = \theta_n + \alpha_{n+1} G_{n+1} f(\theta_n, \xi_{n+1}) \qquad G_{n+1} = -[\widehat{A}_{n+1}]^{-1}$$

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

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

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

Zap Q-Learning: $f(\theta_n, \xi_{n+1}) = \{c(X_n, U_n) + \gamma \underline{Q}^{\theta^*}(X_{n+1}) - Q^{\theta^*}(X_n, U_n)\}\zeta_n$

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

$$A_{n+1} = \zeta_n \big[\gamma\psi(X_{n+1}, \phi^\theta(X_{n+1})) - \psi(X_n, U_n)\big]^T$$

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

Challenges

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

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

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

Challenges

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

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

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

$\Longrightarrow$ 3 resolved for Zap by exploiting special structure, even for NN function approximation [38, 8]

Challenges

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

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

What makes theory difficult:

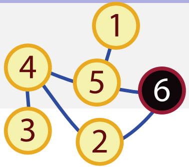
1. Does $\bar{f}$ have a root?
2. Does the inverse of A exist?
3. SA theory is weak for a discontinuous ODE

$\Longrightarrow$ 3 resolved for Zap by exploiting special structure,
even for NN function approximation [38, 8]

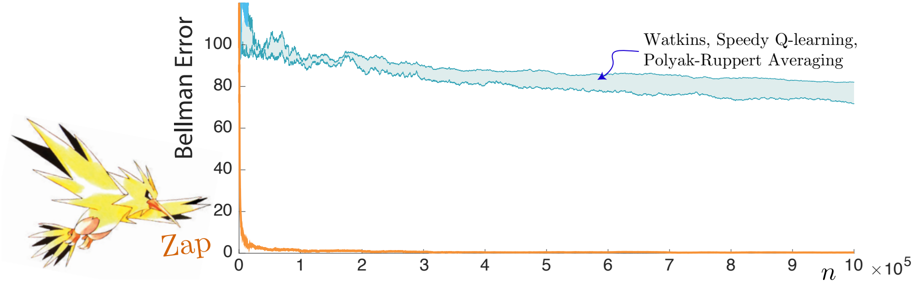
Conclusions for Zap: Stability and optimal asymptotic covariance Σ^*

Zap Q-Learning

Optimize Walk to Cafe



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

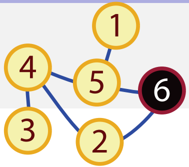


Convergence of Zap-Q Learning

Discount factor: $\gamma = 0.99$

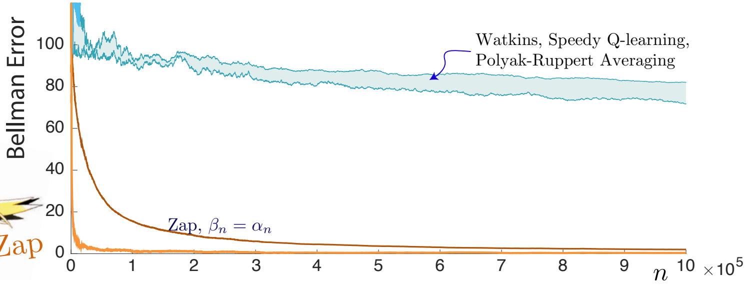
Zap Q-Learning

Optimize Walk to Cafe



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

Infinite covariance with $\alpha_n = 1/n$ or $1/n(x,u)$.

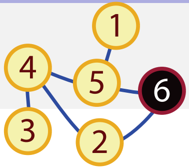


Convergence of Zap-Q Learning

Discount factor: $\gamma = 0.99$

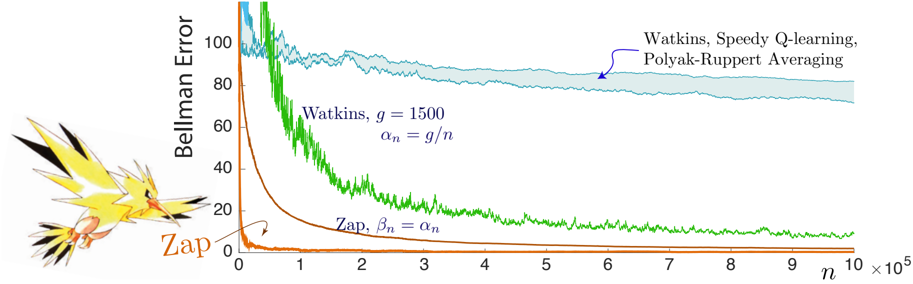
Zap Q-Learning

Optimize Walk to Cafe



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

Infinite covariance with $\alpha_n = 1/n$ or $1/n(x,u)$.

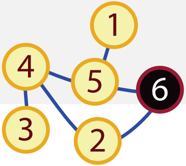


Convergence of Zap-Q Learning

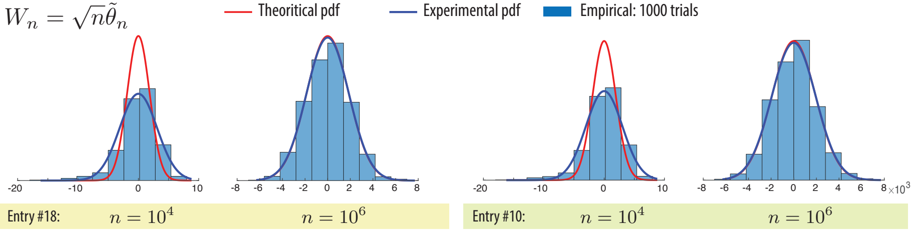
Discount factor: $\gamma = 0.99$

Zap Q-Learning

Optimize Walk to Cafe



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



CLT gives good prediction of finite-n performance

Discount factor: $\gamma = 0.99$

Zap with Neural Networks

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

$$\zeta_n = \nabla_\theta Q^\theta(X_n, U_n)\big|_{\theta=\theta_n}$$ computed using back-progagation

A few things to note:

- As far as we know, the empirical success of plain vanilla DQN is *extraordinary* (however, nobody reports failure)

Zap with Neural Networks

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

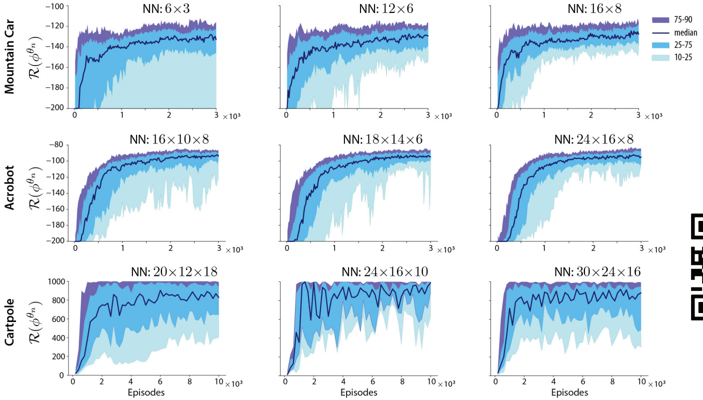
$$\zeta_n = \nabla_\theta Q^\theta(X_n, U_n)\big|_{\theta=\theta_n}$$ computed using back-progagation

A few things to note:

- As far as we know, the empirical success of plain vanilla DQN is *extraordinary* (however, nobody reports failure)
- Zap Q-learning is the only approach for which convergence has been established (under mild conditions)
- We can expect stunning transient performance, based on coupling with the Newton-Raphson flow.

Zap with Neural Networks

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



Reliability and stunning transient performance

—from coupling with the Newton-Raphson flow.

Challenges

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

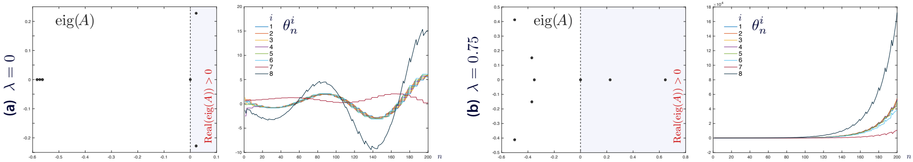
$$\bar{f}(\theta) = \mathsf{E}\big[\{c(X_n,U_n) + \gamma \underline{Q}^\theta(X_{n+1}) - Q^\theta(X_n,U_n)\}\zeta_n\big]$$

What makes theory difficult:

1. Does $\bar{f}$ have a root?
2. Does the inverse of A exist?

$$A(\theta) = \partial_\theta \bar{f}\,(\theta)$$

The Projected Bellman Equation



Challenges

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

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

What makes theory difficult:

1. Does $\bar{f}$ have a root?
2. Does the inverse of A exist?

Stability & The Projected Bellman Equation

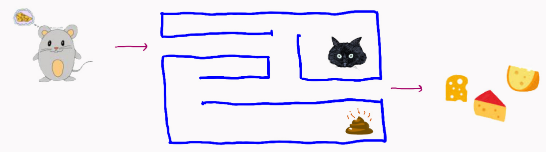
Theory and Practice

Most of the elegant theory for tabular Q-learning: training is *oblivious*

Theory and Practice

Most of the elegant theory for tabular Q-learning: training is *oblivious*

In practice we follow the intelligent mouse



I only need to see the cat *once*

Theory and Practice

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

Most of the elegant theory for tabular Q-learning: training is *oblivious*

In practice we follow the intelligent mouse

Approaches to exploration, $U_k \sim \breve{\phi}_k(\,\cdot \mid X_k)$:

- ε-greedy, $U_k = \phi^\theta(X_k)$ probability $1-\varepsilon$ small $\varepsilon > 0$
- Gibbs, $\breve{\phi}_k(u \mid x) = \dfrac{1}{\mathcal{Z}} \exp\bigl(-\kappa Q^{\theta_k}(x,u)\bigr)$ large $\kappa > 0$

Theory and Practice

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

Most of the elegant theory for tabular Q-learning: training is *oblivious*

In practice we follow the intelligent mouse

Approaches to exploration, $U_k \sim \breve{\phi}_k(\,\cdot \mid X_k)$:

- ε-greedy, $U_k = \phi^{\theta}(X_k)$ probability $1-\varepsilon$ — small $\varepsilon > 0$

  Discontinuous vector field

- Gibbs, $\breve{\phi}_k(u \mid x) = \dfrac{1}{\mathcal{Z}} \exp\bigl(-\kappa Q^{\theta_k}(x,u)\bigr)$ — large $\kappa > 0$

  Lipschitz fails (and more)

Theory and Practice

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

Most of the elegant theory for tabular Q-learning: training is *oblivious*

In practice we follow the intelligent mouse

Approaches to exploration, $U_k \sim \check{\phi}_k(\,\cdot \mid X_k)$:

- ε-greedy, $U_k = \phi^\theta(X_k)$ probability $1-\varepsilon$ — small $\varepsilon > 0$

  Discontinuous vector field

- Gibbs, $\check{\phi}_k(u \mid x) = \dfrac{1}{\mathcal{Z}} \exp\bigl(-\kappa Q^{\theta_k}(x,u)\bigr)$ — large $\kappa > 0$

  Lipschitz fails (and more)

  Approximates ε-greedy policy with $\varepsilon = 0$ if $\|\theta_k\|$ is large

Theory and Practice

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

Most of the elegant theory for tabular Q-learning: training is *oblivious*

In practice we follow the intelligent mouse

Approaches to exploration, $U_k \sim \breve{\phi}_k(\,\cdot \mid X_k)$:

- ε-greedy, $U_k = \phi^\theta(X_k)$ probability $1-\varepsilon$ — small $\varepsilon > 0$
- Gibbs, $\breve{\phi}_k(u \mid x) = \frac{1}{\mathcal{Z}} \exp\bigl(-\kappa Q^{\theta_k}(x,u)\bigr)$ — large $\kappa > 0$
- Tamed Gibbs, $\breve{\phi}^\theta_0(u \mid x) = \frac{1}{\mathcal{Z}^\theta_\kappa(x)} \exp\bigl(-\kappa_\theta Q^\theta(x,u)\bigr)$ — New in 2023

Theory and Practice

$$\phi^{\theta}(x) = \arg\min_{u} Q^{\theta}(x,u)$$

Most of the elegant theory for tabular Q-learning: training is *oblivious*

In practice we follow the intelligent mouse

Approaches to exploration, $U_k \sim \breve{\phi}_k(\,\cdot \mid X_k)$:

- ε-greedy, $U_k = \phi^{\theta}(X_k)$ probability $1-\varepsilon$ — small $\varepsilon > 0$

- Gibbs, $\breve{\phi}_k(u \mid x) = \dfrac{1}{\mathcal{Z}} \exp\bigl(-\kappa Q^{\theta_k}(x,u)\bigr)$ — large $\kappa > 0$

- Tamed Gibbs, $\breve{\phi}^{\theta}_0(u \mid x) = \dfrac{1}{\mathcal{Z}^{\theta}_{\kappa}(x)} \exp\bigl(-\kappa_\theta Q^{\theta}(x,u)\bigr)$ — large $\kappa_0 > 0$

$$\kappa_\theta \begin{cases} = \frac{1}{\|\theta\|}\kappa_0 & \|\theta\| \ge 1 \\ \ge \frac{1}{2}\kappa_0 & \textit{else} \end{cases}$$

SA recursion satisfies all the assumptions 🙂 New in 2023

Theory for Tamed Gibbs $\breve{\phi}_k(u \mid x) \stackrel{\text{def}}{=} \mathsf{P}\{U_k = u \mid \mathcal{F}_k \,; X_k = x\}$

For ease of analysis: $\breve{\phi}_k(u \mid x) = (1-\varepsilon)\breve{\phi}_0^{\theta_k}(u \mid x) + \varepsilon \nu_{\mathcal{W}}(u)$

Theory for Tamed Gibbs $\breve{\phi}_k(u \mid x) \stackrel{\text{def}}{=} \mathsf{P}\{U_k = u \mid \mathcal{F}_k\,; X_k = x\}$

For ease of analysis: $\breve{\phi}_k(u \mid x) = (1-\varepsilon)\breve{\phi}_0^{\theta_k}(u \mid x) + \varepsilon \nu_{\mathcal{W}}(u)$

Assumptions: $Q^\theta(x,u) = \theta^\intercal \psi(x,u)$, and

Theory for Tamed Gibbs $\quad \check{\phi}_k(u \mid x) \stackrel{\text{def}}{=} \mathsf{P}\{U_k = u \mid \mathcal{F}_k \,;\, X_k = x\}$

For ease of analysis: $\check{\phi}_k(u \mid x) = (1-\varepsilon)\check{\phi}_0^{\theta_k}(u \mid x) + \varepsilon \nu_{\mathcal{W}}(u)$

Assumptions: $\quad Q^\theta(x,u) = \theta^\intercal \psi(x,u)$, and

For oblivious policy ($\varepsilon = 1$):

1. There is a unique invariant pmf $\pi_{\mathcal{W}}$ for $(\boldsymbol{X}, \boldsymbol{U})$.
2. The covariance is full rank, $R^{\mathcal{W}} > 0$,

$$R^{\mathcal{W}} = \mathsf{E}_{\pi_{\mathcal{W}}}\left[\psi(X_n,U_n)\psi(X_n,U_n)^\intercal\right], \qquad U_n = \mathcal{W}_n \sim \nu_{\mathcal{W}}$$

Theory for Tamed Gibbs $\quad \breve{\phi}_k(u \mid x) \stackrel{\text{def}}{=} \mathsf{P}\{U_k = u \mid \mathcal{F}_k \,; X_k = x\}$

For ease of analysis: $\breve{\phi}_k(u \mid x) = (1-\varepsilon)\breve{\phi}_0^{\theta_k}(u \mid x) + \varepsilon \nu_{\mathcal{W}}(u)$

Assumptions: $\quad Q^\theta(x,u) = \theta^{\intercal}\psi(x,u)$, and

For oblivious policy ($\varepsilon = 1$):

1. There is a unique invariant pmf $\pi_{\mathcal{W}}$ for $(\boldsymbol{X}, \boldsymbol{U})$.
2. The covariance is full rank,

$$R^{\mathcal{W}} = \mathsf{E}_{\pi_{\mathcal{W}}}\big[\psi(X_n,U_n)\psi(X_n,U_n)^{\intercal}\big]\,, \qquad U_n = \mathcal{W}_n \sim \nu_{\mathcal{W}}$$

First step in analysis is to show that 1 and 2 hold for any $\varepsilon > 0$:

Theory for Tamed Gibbs $\quad \breve{\phi}_k(u \mid x) \stackrel{\text{def}}{=} \mathsf{P}\{U_k = u \mid \mathcal{F}_k \,;\, X_k = x\}$

For ease of analysis: $\breve{\phi}_k(u \mid x) = (1-\varepsilon)\breve{\phi}_0^{\theta_k}(u \mid x) + \varepsilon \nu_{\mathcal{W}}(u)$

Assumptions: $\quad Q^\theta(x,u) = \theta^{\intercal}\psi(x,u)$, and

For oblivious policy ($\varepsilon = 1$):

1. There is a unique invariant pmf $\pi_{\mathcal{W}}$ for $(\boldsymbol{X}, \boldsymbol{U})$.
2. The covariance is full rank,

$$R^{\mathcal{W}} = \mathsf{E}_{\pi_{\mathcal{W}}}\big[\psi(X_n,U_n)\psi(X_n,U_n)^{\intercal}\big]\,, \qquad U_n = \mathcal{W}_n \sim \nu_{\mathcal{W}}$$

First step in analysis is to show that 1 and 2 hold for any $\varepsilon > 0$:

- There is a unique invariant pmf π_θ for $(\boldsymbol{X}, \boldsymbol{U})$.
- The covariance is full rank,

$$R^{\Theta}(\theta) = \mathsf{E}_{\pi_\theta}\big[\psi(X_n,U_n)\psi(X_n,U_n)^{\intercal}\big]\,, \qquad U_n \sim \breve{\phi}_n(\,\cdot \mid X_n)$$

Theory

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

Stability with sufficient optimism.

There is $\varepsilon_\gamma > 0$ (lower bound given in paper) for which the following hold:

Theory

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

Stability with sufficient optimism.
There is $\varepsilon_\gamma > 0$ (lower bound given in paper) for which the following hold:

For each $0 < \varepsilon < \varepsilon_\gamma$, there is $\kappa_{\varepsilon,\gamma}$ such that

- The mean flow $\frac{d}{dt}\vartheta = \bar{f}(\vartheta)$ is *ultimately bounded.*

Proof follows Van Roy's analysis of TD-learning,

$$\frac{d}{dt}\|\vartheta_t\| \le -\delta\|\vartheta_t\|\,, \qquad \textit{if}\ \ \|\vartheta_t\| \ge 1/\delta$$

Theory

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

Stability with sufficient optimism.

There is $\varepsilon_\gamma > 0$ (lower bound given in paper) for which the following hold:

For each $0 < \varepsilon < \varepsilon_\gamma$, there is $\kappa_{\varepsilon,\gamma}$ such that

- The mean flow $\frac{d}{dt}\vartheta = \bar{f}(\vartheta)$ is *ultimately bounded*.
- There is at least one solution to the projected Bellman equation

$$\bar{f}(\theta^*) = 0$$

Existence of θ^* follows from the stability proof:

Denote $T(\theta) = \theta + \varepsilon_0 \bar{f}(\theta)$ for $\theta \in \mathbb{R}^d$, with $\varepsilon_0 > 0$ sufficiently small.

Goal: solve $T(\theta^*) = \theta^*$

Theory

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

Stability with sufficient optimism.
There is $\varepsilon_\gamma > 0$ (lower bound given in paper) for which the following hold:

For each $0 < \varepsilon < \varepsilon_\gamma$, there is $\kappa_{\varepsilon,\gamma}$ such that

- The mean flow $\frac{d}{dt}\vartheta = \bar{f}(\vartheta)$ is *ultimately bounded*.
- There is at least one solution to the projected Bellman equation

$$\bar{f}(\theta^*) = 0$$

Existence of θ^* follows from the stability proof:

Denote $T(\theta) = \theta + \varepsilon_0 \bar{f}(\theta)$ for $\theta \in \mathbb{R}^d$, with $\varepsilon_0 > 0$ sufficiently small.

$$\|T(\theta)\| \le 1/\delta\,, \qquad \textit{if} \;\; \|\theta\| \le 1/\delta \qquad \textit{for some } \delta > 0$$

Brouwer's fixed-point theorem tells us $T(\theta^*) = \theta^*$ has at least one solution.

See also de Farias & Van Roy [11]

Theory

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

Stability with sufficient optimism.
There is $\varepsilon_\gamma > 0$ (lower bound given in paper) for which the following hold:

For each $0 < \varepsilon < \varepsilon_\gamma$, there is $\kappa_{\varepsilon,\gamma}$ such that

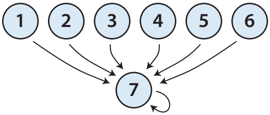
- The mean flow $\frac{d}{dt}\vartheta = \bar{f}(\vartheta)$ is *ultimately bounded*.
- There is at least one solution to the projected Bellman equation

$$\bar{f}(\theta^*) = 0$$

- Under some additional assumptions θ^* is *locally* asymptotically stable

Baird's Example

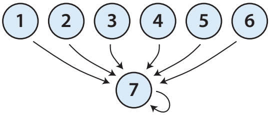
$$\breve{\phi}_k(u \mid x) = (1-\varepsilon)\breve{\phi}_0^{\theta_k}(u \mid x) + \varepsilon \nu_{\mathcal{W}}(u)$$



$$h^\theta(x) = \theta^{\intercal}\psi(x) = \begin{cases} \theta^8 + 2\theta^k & x = k \le 6 \\ 2\theta^8 + \theta^7 & x = 7 \end{cases}$$

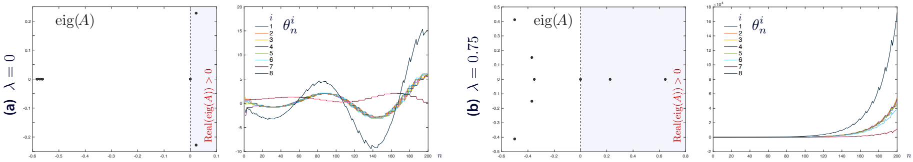
Baird's Example

$$\check{\phi}_k(u \mid x) = (1-\varepsilon)\check{\phi}_0^{\theta_k}(u \mid x) + \varepsilon \nu_{\mathcal{W}}(u)$$



$$h^\theta(x) = \theta^T \psi(x) = \begin{cases} \theta^8 + 2\theta^k & x = k \le 6 \\ 2\theta^8 + \theta^7 & x = 7 \end{cases}$$

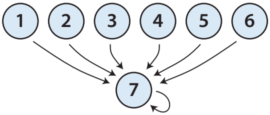
The need for $\varepsilon > 0$ sufficiently small:



From CS&RL Results for TD(λ)-learning, $\varepsilon = 1$

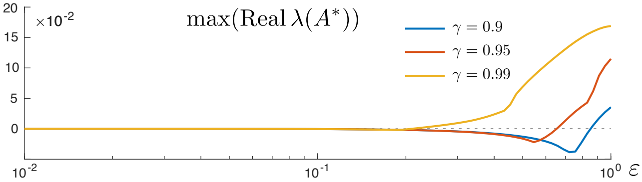
Baird's Example

$$\breve{\phi}_k(u \mid x) = (1-\varepsilon)\breve{\phi}_0^{\theta_k}(u \mid x) + \varepsilon \nu_{\mathcal{W}}(u)$$



$$h^\theta(x) = \theta^{\mathsf{T}}\psi(x) = \begin{cases} \theta^8 + 2\theta^k & x = k \le 6 \\ 2\theta^8 + \theta^7 & x = 7 \end{cases}$$

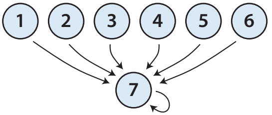
The need for $\varepsilon > 0$ sufficiently small:



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

Baird's Example

$$\check{\phi}_k(u \mid x) = (1-\varepsilon)\check{\phi}_0^{\theta_k}(u \mid x) + \varepsilon \nu_{\mathcal{W}}(u)$$



$$h^\theta(x) = \theta^T \psi(x) = \begin{cases} \theta^8 + 2\theta^k & x = k \le 6 \\ 2\theta^8 + \theta^7 & x = 7 \end{cases}$$

The need for $\varepsilon > 0$ sufficiently small:

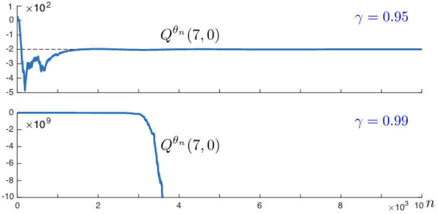
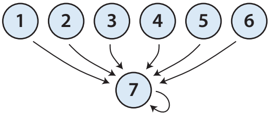


Fig. 1. Evolution of the Q-function approximations for two values of discount factor, and using an ε-greedy policy with common value of $\varepsilon = 0.5$.

Baird's Example

$$\check{\phi}_k(u \mid x) = (1-\varepsilon)\check{\phi}_0^{\theta_k}(u \mid x) + \varepsilon \nu_{\mathcal{W}}(u)$$



$$h^\theta(x) = \theta^T \psi(x) = \begin{cases} \theta^8 + 2\theta^k & x = k \le 6 \\ 2\theta^8 + \theta^7 & x = 7 \end{cases}$$

The need for $\varepsilon > 0$ sufficiently small:

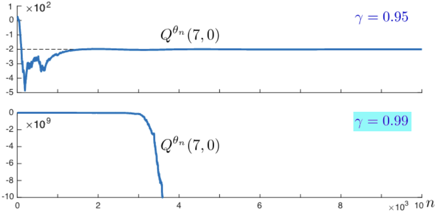


Fig. 1. Evolution of the Q-function approximations for two values of discount factor, and using an ε-greedy policy with common value of $\varepsilon = 0.5$.

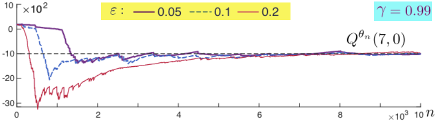
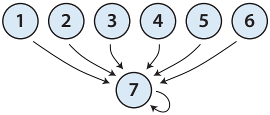


Fig. 2. Evolution of the Q-function approximations when using an ε-greedy policy. Convergence holds when $\varepsilon > 0$ is sufficiently small.

Baird's Example

$$\check{\phi}_k(u \mid x) = (1-\varepsilon)\check{\phi}_0^{\theta_k}(u \mid x) + \varepsilon \nu_{\mathcal{W}}(u)$$



$$h^\theta(x) = \theta^T \psi(x) = \begin{cases} \theta^8 + 2\theta^k & x = k \le 6 \\ 2\theta^8 + \theta^7 & x = 7 \end{cases}$$

The need for $\varepsilon > 0$ sufficiently small:

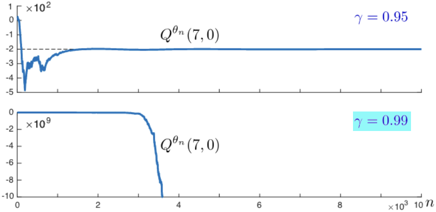


Fig. 1. Evolution of the Q-function approximations for two values of discount factor, and using an ε-greedy policy with common value of $\varepsilon = 0.5$.

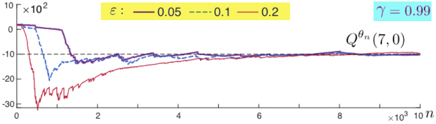


Fig. 2. Evolution of the Q-function approximations when using an ε-greedy policy. Convergence holds when $\varepsilon > 0$ is sufficiently small.

Recent application to change detection, using Zap: $A^* = \partial_\theta \bar{f}(\theta^*)$ is not Hurwitz [82].

Thoughts on Implementation

- Use of relative Q-learning or advantage Q-learning can improve numerical stability: estimate $H^*(x,u) = Q^*(x,u) - \xi(x)$ for appropriate function ξ.

Thoughts on Implementation

- Use of relative Q-learning or advantage Q-learning can improve numerical stability: estimate $H^*(x,u) = Q^*(x,u) - \xi(x)$ for appropriate function ξ.

- Theory extends to ε-optimistic SARSA:

$$\theta_{n+1} = \theta_n + \alpha_{n+1}\big\{c(X_n,U_n) + \gamma Q^{\theta_n}(X_{n+1},U_{n+1}) - Q^{\theta_n}(X_n,U_n)\big\}\zeta_n$$

with $\{U_n\}$ defined by your favorite variant of the ε-greedy policy.

Thoughts on Implementation

• Use of relative Q-learning or advantage Q-learning can improve numerical stability: estimate $H^*(x,u) = Q^*(x,u) - \xi(x)$ for appropriate function ξ.

• Theory extends to ε-optimistic SARSA:

$$\theta_{n+1} = \theta_n + \alpha_{n+1}\{c(X_n, U_n) + \gamma Q^{\theta_n}(X_{n+1}, U_{n+1}) - Q^{\theta_n}(X_n, U_n)\}\zeta_n$$

with $\{U_n\}$ defined by your favorite variant of the ε-greedy policy.

• However, remember that to-date we only have the existence of θ^* and ultimate boundedness of $\{\theta_n\}$, provided $\varepsilon > 0$.

Convergence remains an open topic for research

Conclusions & Future Directions

- Reinforcement Learning is cursed by dimension, variance, and nonlinear (algorithm) dynamics (far more than a triad)

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

- 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 [67, 68, 69, 70]

- 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 [67, 68, 69, 70]
- Zap with optimism:

$$
\begin{aligned}
A(\theta) &= \partial_\theta \mathsf{E}_{\pi_\theta}[f(\theta, \Xi_n)] \\
&= \mathsf{E}_{\pi_\theta}[\partial_\theta f(\theta, \Xi_n)] + \mathsf{E}_{\pi_\theta}[\hat{f}(\theta, \Xi_n)\Lambda_\theta(\Xi_n)^{\intercal}]
\end{aligned}
$$

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

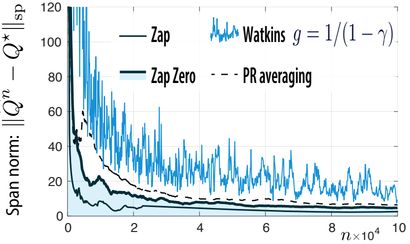
Future work:

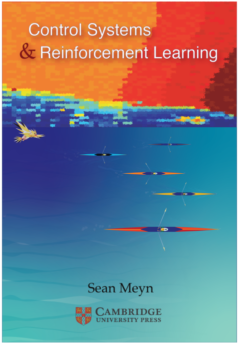
- Beyond the projected Bellman error for Q-learning [67, 68, 69, 70]
- Zap with optimism
- Acceleration techniques (momentum and matrix momentum)
  See Zap-Zero in CS&RL (and big improvements in [10])

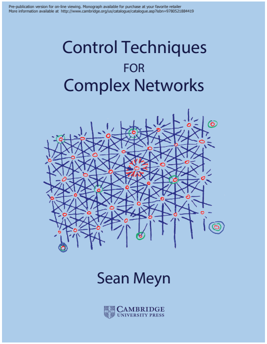
- Reinforcement Learning is cursed by dimension, variance, and nonlinear (algorithm) dynamics
- Second order methods can ensure stability—use them when you can

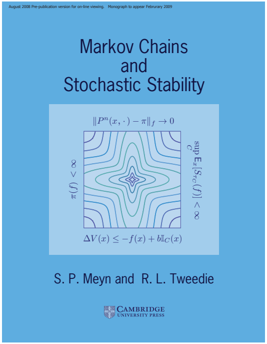
Future work:

- Beyond the projected Bellman error for Q-learning [67, 68, 69, 70]
- Zap with optimism
- Acceleration techniques (momentum and matrix momentum)
  See Zap-Zero in CS&RL (and big improvements in [10]):




Control Systems
& Reinforcement Learning
Sean Meyn
CAMBRIDGE
UNIVERSITY PRESS


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

References

Control Background I

[1] K. J. Åström and B. Wittenmark. *Adaptive Control*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 1994.

[2] A. Fradkov and B. T. Polyak. Adaptive and robust control in the USSR. *IFAC–PapersOnLine*, 53(2):1373–1378, 2020. 21th IFAC World Congress.

[3] M. Krstic, P. V. Kokotovic, and I. Kanellakopoulos. *Nonlinear and adaptive control design*. John Wiley & Sons, Inc., 1995.

[4] K. J. Åström. Theory and applications of adaptive control—a survey. *Automatica*, 19(5):471–486, 1983.

[5] K. J. Åström. Adaptive control around 1960. *IEEE Control Systems Magazine*, 16(3):44–49, 1996.

[6] L. Ljung. Analysis of recursive stochastic algorithms. *IEEE Transactions on Automatic Control*, 22(4):551–575, 1977.

[7] N. Matni, A. Proutiere, A. Rantzer, and S. Tu. From self-tuning regulators to reinforcement learning and back again. In *Proc. of the IEEE Conf. on Dec. and Control*, pages 3724–3740, 2019.

RL Background I

[8] S. Meyn. *Control Systems and Reinforcement Learning*. Cambridge University Press, Cambridge, 2021.

[9] S. Meyn. *The projected Bellman equation in reinforcement learning. IEEE Transactions on Automatic Control*, pages 8323–8337, 2024.

[10] S. Meyn. *Stability of Q-learning through design and optimism. arXiv 2307.02632*, 2023.

[11] D. De Farias and B. Van Roy. On the existence of fixed points for approximate value iteration and temporal-difference learning. *Journal of Optimization Theory and Applications*, 105(3):589–608, 2000.

[12] Z. Chen, J.-P. Clarke, and S. T. Maguluri. Target network and truncation overcome the deadly triad in Q-learning. *SIAM Journal on Mathematics of Data Science*, 5(4):1078–1101, 2023.

[13] D. Lee and N. He. A unified switching system perspective and convergence analysis of Q-learning algorithms. *Advances in Neural Information Processing Systems*, 33:15556–15567, 2020.

[14] F. S. Melo, S. P. Meyn, and M. I. Ribeiro. An analysis of reinforcement learning with function approximation. In *Proc. ICML*, pages 664–671, New York, NY, 2008.

RL Background II

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

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

[17] D. P. Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific, Belmont, MA, 2019.

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

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

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

[21] J. Tsitsiklis. Asynchronous stochastic approximation and Q-learning. *Machine Learning*, 16:185–202, 1994.

[22] S. P. Singh, T. Jaakkola, and M. Jordan. Reinforcement learning with soft state aggregation. *Proc. Advances in Neural Information Processing Systems*, 7:361, 1995.

RL Background III

[23] B. Van Roy. *Learning and Value Function Approximation in Complex Decision Processes*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1998.

[24] J. N. Tsitsiklis and B. Van Roy. Feature-based methods for large scale dynamic programming. *Mach. Learn.*, 22(1-3):59–94, 1996.

[25] 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.

[26] J. N. Tsitsiklis and B. V. Roy. Average cost temporal-difference learning. *Automatica*, 35(11):1799–1808, 1999.

[27] 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.

[28] 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.

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

RL Background IV

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

[31] 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.

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

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

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

[35] H. R. Maei, C. Szepesvári, S. Bhatnagar, and R. S. Sutton. Toward off-policy learning control with function approximation. In *Proc. ICML*, pages 719–726, USA, 2010. Omnipress.

[36] 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.

RL Background V

[37] A. M. Devraj and S. P. Meyn. Zap Q-learning. In *Proc. of the Intl. Conference on Neural Information Processing Systems*, pages 2232–2241, 2017.

[38] S. Chen, A. M. Devraj, F. Lu, A. Busic, and S. Meyn. Zap Q-Learning with nonlinear function approximation. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems, and arXiv e-prints 1910.05405*, volume 33, pages 16879–16890, 2020.

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

DQN:

[40] M. Riedmiller. *Neural fitted Q iteration – first experiences with a data efficient neural reinforcement learning method.* In J. Gama, R. Camacho, P. B. Brazdil, A. M. Jorge, and L. Torgo, editors, *Machine Learning: ECML 2005*, pages 317–328, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[41] S. Lange, T. Gabel, and M. Riedmiller. *Batch reinforcement learning.* In *Reinforcement learning*, pages 45–73. Springer, 2012.

[42] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller. *Playing Atari with deep reinforcement learning. ArXiv*, abs/1312.5602, 2013.

RL Background VI

[43] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. A. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. *Human-level control through deep reinforcement learning.* *Nature*, 518:529–533, 2015.

[44] O. Anschel, N. Baram, and N. Shimkin. Averaged-DQN: Variance reduction and stabilization for deep reinforcement learning. In *Proc. of ICML*, pages 176–185. JMLR.org, 2017.

Actor Critic / Policy Gradient

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

[46] C. D. Meyer, Jr. The role of the group generalized inverse in the theory of finite Markov chains. *SIAM Review*, 17(3):443–464, 1975.

[47] P. W. Glynn. Stochastic approximation for Monte Carlo optimization. In *Proceedings of the 18th conference on Winter simulation*, pages 356–365, 1986.

[48] P. W. Glynn. Likelihood ratio gradient estimation: An overview. In *Proc. of the Winter Simulation Conference*, pages 366–375, 1987.

RL Background VII

[49] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[50] J. C. Spall. A one-measurement form of simultaneous perturbation stochastic approximation. *Automatica*, 33(1):109–112, 1997.

[51] T. Jaakkola, S. P. Singh, and M. I. Jordan. Reinforcement learning algorithm for partially observable Markov decision problems. In *Advances in neural information processing systems*, pages 345–352, 1995.

[52] X.-R. Cao and H.-F. Chen. Perturbation realization, potentials, and sensitivity analysis of Markov processes. *IEEE Transactions on Automatic Control*, 42(10):1382–1393, Oct 1997.

[53] P. Marbach and J. N. Tsitsiklis. Simulation-based optimization of Markov reward processes. *IEEE Trans. Automat. Control*, 46(2):191–209, 2001.

[54] V. Konda. *Actor-critic algorithms*. PhD thesis, Massachusetts Institute of Technology, 2002.

[55] V. R. Konda and J. N. Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.

RL Background VIII

[56] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

[57] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

[58] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[59] P. Marbach and J. N. Tsitsiklis. Simulation-based optimization of Markov reward processes. *IEEE Trans. Automat. Control*, 46(2):191–209, 2001.

[60] S. M. Kakade. A natural policy gradient. In *Advances in neural information processing systems*, pages 1531–1538, 2002.

[61] H. Mania, A. Guy, and B. Recht. Simple random search provides a competitive approach to reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1800–1809, 2018.

MDPs, LPs and Convex Q:

RL Background IX

[62] A. S. Manne. Linear programming and sequential decisions. *Management Sci.*, 6(3):259–267, 1960.

[63] C. Derman. *Finite State Markovian Decision Processes*, volume 67 of *Mathematics in Science and Engineering*. Academic Press, Inc., 1970.

[64] V. S. Borkar. Convex analytic methods in Markov decision processes. In *Handbook of Markov decision processes*, volume 40 of *Internat. Ser. Oper. Res. Management Sci.*, pages 347–375. Kluwer Acad. Publ., Boston, MA, 2002.

[65] D. P. de Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations Res.*, 51(6):850–865, 2003.

[66] D. P. de Farias and B. Van Roy. *A cost-shaping linear program for average-cost approximate dynamic programming with performance guarantees. Math. Oper. Res.*, 31(3):597–620, 2006.

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

[68] J. Bas Serrano, S. Curi, A. Krause, and G. Neu. Logistic Q-learning. In A. Banerjee and K. Fukumizu, editors, *Proc. of The Intl. Conference on Artificial Intelligence and Statistics*, volume 130, pages 3610–3618, 13–15 Apr 2021.

RL Background X

[69] F. Lu, P. G. Mehta, S. P. Meyn, and G. Neu. Convex Q-learning. In *American Control Conf.*, pages 4749–4756. IEEE, 2021.

[70] F. Lu, P. G. Mehta, S. P. Meyn, and G. Neu. Convex analytic theory for convex Q-learning. In *IEEE Conference on Decision and Control*, pages 4065–4071, Dec 2022.

Gator Nation:

[71] A. M. Devraj, A. Bušić, and S. Meyn. Fundamental design principles for reinforcement learning algorithms. In K. G. Vamvoudakis, Y. Wan, F. L. Lewis, and D. Cansever, editors, *Handbook on Reinforcement Learning and Control*, Studies in Systems, Decision and Control series (SSDC, volume 325). Springer, 2021.

[72] A. M. Devraj and S. P. Meyn. *Fastest convergence for Q-learning. ArXiv* , July 2017 (extended version of NIPS 2017).

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

[74] A. M. Devraj and S. P. Meyn. Q-learning with uniformly bounded variance: Large discounting is not a barrier to fast learning. *IEEE Trans Auto Control (and arXiv:2002.10301)*, 2021.

RL Background XI

[75] A. M. Devraj and S. P. Meyn. Q-learning with uniformly bounded variance: Large discounting is not a barrier to fast learning. *arXiv:2002.10301 (extended version of IEEE Trans Auto Control, 2022)*, 2021.

[76] A. M. Devraj and S. P. Meyn.
Q-learning with uniformly bounded variance.
IEEE Trans. on Automatic Control, 67(11):5948–5963, 2022.

[77] 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.

[78] C. K. Lauand and S. Meyn. Extremely fast convergence rates for extremum seeking control with Polyak-Ruppert averaging. *arXiv 2206.00814*, 2022.

[79] C. K. Lauand and S. Meyn. Quasi-stochastic approximation: Design principles with applications to extremum seeking control. *IEEE Control Systems Magazine*, 43(5):111–136, Oct 2023.

[80] C. K. Lauand and S. Meyn. Markovian foundations for quasi stochastic approximation. *SIAM Journal on Control and Optimization (pre-publication version arXiv 2207.06371)*, 63(1):402–430, 2025.

RL Background XII

[81] C. K. Lauand and S. Meyn. Revisiting step-size assumptions in stochastic approximation. *arXiv 2405.17834*, 2024.

[82] A. Cooper and S. Meyn. Reinforcement learning design for quickest change detection—extended paper. *arXiv:2403.14109*, 2024.

Stochastic Miscellanea I

[83] S. Asmussen and P. W. Glynn. *Stochastic Simulation: Algorithms and Analysis*, volume 57 of *Stochastic Modelling and Applied Probability*. Springer-Verlag, New York, 2007.

[84] P. W. Glynn and S. P. Meyn. *A Liapounov bound for solutions of the Poisson equation. Ann. Probab.*, 24(2):916–931, 1996.

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

[86] R. Douc, E. Moulines, P. Priouret, and P. Soulier. *Markov Chains*. Springer, 2018.

Stochastic Approximation I

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

[88] A. Benveniste, M. Métivier, and P. Priouret. *Adaptive algorithms and stochastic approximations*, volume 22 of *Applications of Mathematics (New York)*. Springer-Verlag, Berlin, 1990. Translated from the French by Stephen S. Wilson.

[89] 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.

[90] V. Borkar, S. Chen, A. Devraj, I. Kontoyiannis, and S. Meyn. The ODE method for asymptotic statistics in stochastic approximation and reinforcement learning. To appear, Annals of Applied Probability (preprint at arXiv e-prints:2110.14427), 2021.

[91] A. Durmus, E. Moulines, A. Naumov, and S. Samsonov. Finite-time high-probability bounds for Polyak–Ruppert averaged iterates of linear stochastic approximation. *Mathematics of Operations Research*, 2024.

[92] C. K. Lauand and S. Meyn. Revisiting step-size assumptions in stochastic approximation. *arXiv 2405.17834*, 2024.

Stochastic Approximation II

[93] C. K. Lauand and S. Meyn. The curse of memory in stochastic approximation. In *Proc. IEEE Conference on Decision and Control (extended version available at arXiv 2309.02944)*, pages 7803–7809, 2023.

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

[95] V. Borkar and S. P. Meyn. Oja's algorithm for graph clustering, Markov spectral decomposition, and risk sensitive control. *Automatica*, 48(10):2512–2519, 2012.

[96] J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *Ann. Math. Statist.*, 23(3):462–466, 09 1952.

[97] J. C. Spall. *Introduction to stochastic search and optimization: estimation, simulation, and control.* John Wiley & Sons, 2003.

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

[99] 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.

Stochastic Approximation III

[100] 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*.

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

[102] V. R. Konda and V. S. Borkar. Actor-critic–type learning algorithms for Markov decision processes. *SIAM Journal on control and Optimization*, 38(1):94–123, 1999.

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

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

[105] W. Mou, C. Junchi Li, M. J. Wainwright, P. L. Bartlett, and M. I. Jordan. On Linear Stochastic Approximation: Fine-grained Polyak-Ruppert and Non-Asymptotic Concentration. *arXiv e-prints*, page arXiv:2004.04719, Apr. 2020.

Optimization and ODEs I

[106] W. Su, S. Boyd, and E. Candes. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. In *Advances in neural information processing systems*, pages 2510–2518, 2014.

[107] B. Shi, S. S. Du, W. Su, and M. I. Jordan. Acceleration via symplectic discretization of high-resolution differential equations. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5744–5752. Curran Associates, Inc., 2019.

[108] B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[109] Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. In *Soviet Mathematics Doklady*, 1983.