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

Part 2: Every Optimization Problem Is a Quadratic Program

and implications for Q Learning

Sean Meyn

Department of Electrical and Computer Engineering — University of Florida

Inria International Chair — Inria, Paris

Thanks to to our sponsors: NSF and ARO

Part 2: From DP to QP to Q

Outline

Optimal Control and RL

From DP to Q-learning

$X_{k+1} = \mathrm{F}(X_k, U_k)$

Value function: $$J^\star(x) = \min_{\boldsymbol{u}} \sum_{k=0}^{\infty} c(X_k, U_k)\,, \quad X_0 = x \in \mathsf{X}$$

DP eqn: $$J^\star(X_k) = \min_{U_k} \{ \underbrace{c(X_k, U_k) + J^\star(X_{k+1})}_{Q^\star(X_k, U_k)} \}$$

A conditional expectation would appear for a Markovian model

From DP to Q-learning

$$X_{k+1} = \mathrm{F}(X_k, U_k)$$

Value function: $$J^\star(x) = \min_{\boldsymbol{u}} \sum_{k=0}^{\infty} c(X_k, U_k)\,, \quad X_0 = x \in \mathsf{X}$$

DP eqn: $$J^\star(X_k) = \min_{U_k}\{\underbrace{c(X_k, U_k) + J^\star(X_{k+1})}_{Q^\star(X_k, U_k)}\}$$

DP for Q: $$Q^\star(X_k, U_k) = c(X_k, U_k) + \underline{Q}^\star(X_{k+1})$$

From DP to Q-learning

$$X_{k+1} = \mathrm{F}(X_k, U_k)$$

Value function: $$J^\star(x) = \min_{\boldsymbol{u}} \sum_{k=0}^{\infty} c(X_k, U_k)\,, \quad X_0 = x \in \mathsf{X}$$

DP eqn: $$J^\star(X_k) = \min_{U_k}\{\underbrace{c(X_k, U_k) + J^\star(X_{k+1})}_{Q^\star(X_k, U_k)}\}$$

DP for Q: $$Q^\star(X_k, U_k) = c(X_k, U_k) + \underline{Q}^\star(X_{k+1})$$

Model Free Error Representation for Bellman Error

$$\mathcal{E}^\theta(X_k, U_k) = -Q^\theta(X_k, U_k) + c(X_k, U_k) + \underline{Q}^\theta(X_{k+1})$$

Find θ^* among family $\{Q^\theta(x,u) : \theta \in \mathbb{R}^d\}$

Temporal Difference Methods

$X_{k+1} = \mathrm{F}(X_k, U_k)$

Dynamic Programming

DP eqn: $$J^\star(x) = \min_{U_k}\{c(x,u) + J^\star(\mathrm{F}(x,u))\}$$

Temporal Difference Methods

$X_{k+1} = \mathrm{F}(X_k, U_k)$

Dynamic Programming

$$\text{DP eqn:} \quad J^\star(x) = \min_{U_k}\{c(x,u) + J^\star(\mathrm{F}(x,u))\}$$

Policy Iteration: Given initial policy ϕ^0: $U_k = \phi^0(X_k)$

1. *Solve* the fixed-policy Bellman equation:

$$J^{\phi^0}(x) = \underbrace{c(x,u) + J^{\phi^0}(\mathrm{F}(x,u))}_{Q^{\phi^0}(x,u)}\Big|_{u=\phi^0(x)}$$

Temporal Difference Methods

$X_{k+1} = \mathrm{F}(X_k, U_k)$

Dynamic Programming

$$\text{DP eqn:} \quad J^\star(x) = \min_{U_k}\{c(x,u) + J^\star(\mathrm{F}(x,u))\}$$

Policy Iteration: Given initial policy ϕ^0: $U_k = \phi^0(X_k)$

1. *Solve* the fixed-policy Bellman equation:

$$J^{\phi^0}(x) = \underbrace{c(x,u) + J^{\phi^0}(\mathrm{F}(x,u))}_{Q^{\phi^0}(x,u)}\Big|_{u=\phi^0(x)}$$

2. Update policy: $\phi^1(x) = \arg\min_u Q^{\phi^0}(x,u)$ *repeat ...*

Temporal Difference Methods

$X_{k+1} = \mathrm{F}(X_k, U_k)$

Dynamic Programming

$$\text{DP eqn:} \quad J^\star(x) = \min_{U_k}\{c(x,u) + J^\star(\mathrm{F}(x,u))\}$$

Policy Iteration: Given initial policy ϕ^0: $U_k = \phi^0(X_k)$

1. *Solve* the fixed-policy Bellman equation:

$$J^{\phi^0}(x) = \underbrace{c(x,u) + J^{\phi^0}(\mathrm{F}(x,u))}_{Q^{\phi^0}(x,u)}\Big|_{u=\phi^0(x)}$$

Fixed policy Bellman equation observed:

$$Q^{\phi^n}(X_k, U_k) = c(X_k, U_k) + Q^{\phi^n}\big(X_{k+1}, \phi^n(X_{k+1})\big)$$

Temporal Difference Methods

$X_{k+1} = \mathrm{F}(X_k, U_k)$

Dynamic Programming

$$\text{DP eqn:} \quad J^\star(x) = \min_{U_k}\{c(x,u) + J^\star(\mathrm{F}(x,u))\}$$

Policy Iteration: Given initial policy ϕ^0: $U_k = \phi^0(X_k)$

1. *Solve* the fixed-policy Bellman equation:

$$J^{\phi^0}(x) = \underbrace{c(x,u) + J^{\phi^0}(\mathrm{F}(x,u))}_{Q^{\phi^0}(x,u)}\Big|_{u=\phi^0(x)}$$

Model Free Error Representation for Bellman Error

$$\mathcal{E}^\theta(X_k, U_k) = -Q^\theta(X_k, U_k) + c(X_k, U_k) + Q^\theta\big(X_{k+1}, \phi^n(X_{k+1})\big)$$

Find θ^* among family $\{Q^\theta(x,u) : \theta \in \mathbb{R}^d\}$

Temporal Difference Methods

$\mathcal{E}^\theta(X_k, U_k) = -Q^\theta(X_k, U_k) + c(X_k, U_k) + \underline{Q}^\theta(X_{k+1})$

Sutton et al recognized the value of the *temporal difference* in the early 80s

TD(λ): estimate value function for fixed policy $U_k = \phi(X_k)$

Modified DP equation: $Q^\phi(X_k, U_k) = c(X_k, U_k) + Q^\phi(X_{k+1}, \phi(X_{k+1}))$

Temporal Difference Methods

$\mathcal{E}^\theta(X_k, U_k) = -Q^\theta(X_k, U_k) + c(X_k, U_k) + \underline{Q}^\theta(X_{k+1})$

Sutton et al recognized the value of the *temporal difference* in the early 80s
TD(λ): estimate value function for fixed policy $U_k = \phi(X_k)$

Modified DP equation: $Q^\phi(X_k, U_k) = c(X_k, U_k) + Q^\phi(X_{k+1}, \phi(X_{k+1}))$
We can keep our definition of $\mathcal{E}^\theta$ with a change of notation:

$$\underline{Q}^\theta(x) = Q^\theta(x, \phi(x))$$

Temporal Difference Methods

$\mathcal{E}^\theta(X_k, U_k) = -Q^\theta(X_k, U_k) + c(X_k, U_k) + \underline{Q}^\theta(X_{k+1})$

Sutton et al recognized the value of the *temporal difference* in the early 80s
TD(λ): estimate value function for fixed policy $U_k = \phi(X_k)$

Modified DP equation: $Q^\phi(X_k, U_k) = c(X_k, U_k) + Q^\phi(X_{k+1}, \phi(X_{k+1}))$
We can keep our definition of $\mathcal{E}^\theta$ with a change of notation:

$$\underline{Q}^\theta(x) = Q^\theta(x, \phi(x))$$

TD(λ) (or SARSA, if you like), attempts to find roots of

$$\bar{f}(\theta) = \mathsf{E}_\infty[\zeta^\theta \mathcal{E}^\theta(X, U)] = \lim_{T\to\infty} \frac{1}{T} \sum_{k=0}^{T-1} \zeta_k^\theta \mathcal{E}^\theta(X_k, U_k)$$

Temporal Difference Methods

$\mathcal{E}^\theta(X_k, U_k) = -Q^\theta(X_k, U_k) + c(X_k, U_k) + \underline{Q}^\theta(X_{k+1})$

Sutton et al recognized the value of the *temporal difference* in the early 80s
TD(λ): estimate value function for fixed policy $U_k = \phi(X_k)$

Choices for the eligibility vector:

$$\text{TD(0)}: \quad \zeta_k^\theta = \nabla_\theta Q^\theta(X_k, U_k)$$

TD(λ) (or SARSA, if you like), attempts to find roots of

$$\bar{f}(\theta) = \mathsf{E}_\infty[\zeta^\theta \mathcal{E}^\theta(X, U)] = \lim_{T\to\infty} \frac{1}{T} \sum_{k=0}^{T-1} \zeta_k^\theta \mathcal{E}^\theta(X_k, U_k)$$

Temporal Difference Methods

$\mathcal{E}^\theta(X_k, U_k) = -Q^\theta(X_k, U_k) + c(X_k, U_k) + \underline{Q}^\theta(X_{k+1})$

Sutton et al recognized the value of the *temporal difference* in the early 80s
TD(λ): estimate value function for fixed policy $U_k = \phi(X_k)$

Choices for the eligibility vector:

$$\text{TD(0):}\quad \zeta_k^\theta = \nabla_\theta Q^\theta(X_k, U_k)$$

$$\text{TD}(\lambda)\text{:}\quad \zeta_k^\theta = \sum_{i=0}^{k} \lambda^i \nabla_\theta Q^\theta(X_{k-i}, U_{k-i})$$

TD(λ) (or SARSA, if you like), attempts to find roots of

$$\bar{f}(\theta) = \mathsf{E}_\infty[\zeta^\theta \mathcal{E}^\theta(X, U)] = \lim_{T\to\infty} \frac{1}{T} \sum_{k=0}^{T-1} \zeta_k^\theta \mathcal{E}^\theta(X_k, U_k)$$

Temporal Difference Methods

$\mathcal{E}^\theta(X_k, U_k) = -Q^\theta(X_k, U_k) + c(X_k, U_k) + \underline{Q}^\theta(X_{k+1})$

$$\bar{f}(\theta) = \mathsf{E}_\infty[\zeta^\theta \mathcal{E}^\theta(X, U)] = \lim_{T\to\infty} \frac{1}{T} \sum_{k=0}^{T-1} \zeta_k^\theta \mathcal{E}^\theta(X_k, U_k)$$

Solution approaches: 1. ODE design: $\frac{d}{dt}\theta_t = G_t \bar{f}(\theta_t)$, and *translation:*

$$\theta_{n+1} = \theta_n + \alpha_{n+1} G_n \zeta_n \mathcal{E}^{\theta_n}(X_n, U_n)$$
$$\zeta_{n+1} = \lambda \zeta_n + \nabla_\theta Q^{\theta_n}(X_{n+1}, U_{n+1})$$

Temporal Difference Methods

$\mathcal{E}^\theta(X_k, U_k) = -Q^\theta(X_k, U_k) + c(X_k, U_k) + \underline{Q}^\theta(X_{k+1})$

$$\bar{f}(\theta) = \mathsf{E}_\infty[\zeta^\theta \mathcal{E}^\theta(X, U)] = \lim_{T\to\infty} \frac{1}{T} \sum_{k=0}^{T-1} \zeta_k^\theta \mathcal{E}^\theta(X_k, U_k)$$

Solution approaches: 1. ODE design: $\frac{d}{dt}\theta_t = G_t \bar{f}(\theta_t)$, and *translation:*

$$\theta_{n+1} = \theta_n + \alpha_{n+1} G_n \zeta_n \mathcal{E}^{\theta_n}(X_n, U_n)$$
$$\zeta_{n+1} = \lambda \zeta_n + \nabla_\theta Q^{\theta_n}(X_{n+1}, U_{n+1})$$

2. LSTD: Consider a linear parameterization $Q^\theta = \theta^\intercal \psi$, giving

$$0 = \frac{1}{T} \sum_{k=0}^{T-1} \zeta_k \mathcal{E}^\theta(X_k, U_k) = A_T \theta - b_T$$

Temporal Difference Methods

$\mathcal{E}^\theta(X_k, U_k) = -Q^\theta(X_k, U_k) + c(X_k, U_k) + \underline{Q}^\theta(X_{k+1})$

$$\bar{f}(\theta) = \mathsf{E}_\infty[\zeta^\theta \mathcal{E}^\theta(X, U)] = \lim_{T\to\infty} \frac{1}{T} \sum_{k=0}^{T-1} \zeta_k^\theta \mathcal{E}^\theta(X_k, U_k)$$

Solution approaches: 1. ODE design: $\frac{d}{dt}\theta_t = G_t \bar{f}(\theta_t)$, and *translation:*

$$\begin{aligned} \theta_{n+1} &= \theta_n + \alpha_{n+1} G_n \zeta_n \mathcal{E}^{\theta_n}(X_n, U_n) \\ \zeta_{n+1} &= \lambda \zeta_n + \nabla_\theta Q^{\theta_n}(X_{n+1}, U_{n+1}) \end{aligned}$$

2. LSTD: Consider a linear parameterization $Q^\theta = \theta^\intercal \psi$, giving

$$0 = \frac{1}{T} \sum_{k=0}^{T-1} \zeta_k \mathcal{E}^\theta(X_k, U_k) = A_T \theta - b_T$$

Amazing fact: $\theta_T^* = A_T^{-1} b_T$ obtained for special gain: $G_n = -A_n^{-1}$

Temporal Difference Methods

Does it work? Let's stick to $Q^\theta = \theta^{\intercal}\psi$

$$\theta_{n+1} = \theta_n + \alpha_{n+1} G_n \zeta_n \mathcal{E}^{\theta_n}(X_n, U_n)$$
$$\zeta_{n+1} = \lambda \zeta_n + \nabla_\theta Q^{\theta_n}(X_{n+1}, U_{n+1})$$

Require exploration, such as $U_k = \widetilde{\phi}(X_k, \xi_k)$ $\Longleftarrow$ QSA theory to come

Persistence of excitation: $$\frac{1}{T}\sum_{k=0}^{T-1} \psi(X_k, U_k)\psi(X_k, U_k)^{\intercal} \to \Sigma_\psi > 0$$

Temporal Difference Methods

Does it work? Let's stick to $Q^\theta = \theta^T\psi$

$$\theta_{n+1} = \theta_n + \alpha_{n+1} G_n \zeta_n \mathcal{E}^{\theta_n}(X_n, U_n)$$
$$\zeta_{n+1} = \lambda\zeta_n + \nabla_\theta Q^{\theta_n}(X_{n+1}, U_{n+1})$$

Require exploration, such as $U_k = \widetilde{\phi}(X_k, \xi_k)$

Persistence of excitation: $\displaystyle \frac{1}{T}\sum_{k=0}^{T-1} \psi(X_k, U_k)\psi(X_k, U_k)^T \to \Sigma_\psi > 0$

Some good news:

- $G_n = A_n^{-1}$ exists! (may fail for at most d values of λ)

Temporal Difference Methods

Does it work? Let's stick to $Q^\theta = \theta^T \psi$

$$\theta_{n+1} = \theta_n + \alpha_{n+1} G_n \zeta_n \mathcal{E}^{\theta_n}(X_n, U_n)$$
$$\zeta_{n+1} = \lambda \zeta_n + \nabla_\theta Q^{\theta_n}(X_{n+1}, U_{n+1})$$

Require exploration, such as $U_k = \widetilde{\phi}(X_k, \xi_k)$

Persistence of excitation: $\displaystyle \frac{1}{T}\sum_{k=0}^{T-1} \psi(X_k, U_k)\psi(X_k, U_k)^T \to \Sigma_\psi > 0$

Some good news:

- $G_n = A_n^{-1}$ exists! (may fail for at most d values of λ)
- TD(1) solves the min-norm problem: $\min_\theta \|Q^\theta - Q^*\|_\pi$

Temporal Difference Methods

Does it work? Let's stick to $Q^\theta = \theta^{\intercal}\psi$

$$\theta_{n+1} = \theta_n + \alpha_{n+1} G_n \zeta_n \mathcal{E}^{\theta_n}(X_n, U_n)$$

Require exploration, such as $U_k = \widetilde{\phi}(X_k, \xi_k)$

Persistence of excitation: $\displaystyle \frac{1}{T}\sum_{k=0}^{T-1} \psi(X_k, U_k)\psi(X_k, U_k)^{\intercal} \to \Sigma_\psi > 0$

Some good news:

- TD(1) solves the min-norm problem: $\min_\theta \|Q^\theta - Q^*\|_\pi$

Well, not so fast!

This beautiful result was obtained for MDPs, in the on-policy setting:

$$U_k = \phi(X_k)$$

π is the steady-state distribution of $\{(X_k, U_k) : k \ge 0\}$

... *do you smell trouble?*

Temporal Difference Methods

Does it work? Let's stick to $Q^\theta = \theta^{\mathsf{T}}\psi$

$$\theta_{n+1} = \theta_n + \alpha_{n+1} G_n \zeta_n \mathcal{E}^{\theta_n}(X_n, U_n)$$

Some good news:

- TD(1) solves the min-norm problem: $\min_\theta \|Q^\theta - Q^*\|_\pi$

This beautiful result was obtained for MDPs, in the on-policy setting:

$$U_k = \phi(X_k)$$

π is the steady-state distribution of $\{(X_k, U_k) : k \geq 0\}$

Potential resolution: on-policy $\oplus$ *re-start* (periodically re-set initial condition)

Temporal Difference Methods

Does it work? Let's stick to $Q^\theta = \theta^T \psi$

$$\theta_{n+1} = \theta_n + \alpha_{n+1} G_n \zeta_n \mathcal{E}^{\theta_n}(X_n, U_n)$$

Some good news:

- TD(1) solves the min-norm problem: $\min_\theta \|Q^\theta - Q^*\|_\pi$

This beautiful result was obtained for MDPs, in the on-policy setting:

$$U_k = \phi(X_k)$$

π is the steady-state distribution of $\{(X_k, U_k) : k \geq 0\}$

Potential resolution: on-policy $\oplus$ *re-start* (periodically re-set initial condition)

However, is minimizing $\|Q^\theta - Q^*\|_\pi$ a compelling goal?

Q(0) Learning and Deep Q-Learning

A generalization of Watkins' algorithm [13, 26, 10]

Model Free Error Representation:

$$\mathcal{E}^\theta(X_k, U_k) = -Q^\theta(X_k, U_k) + c(X_k, U_k) + \underline{Q}^\theta(X_{k+1})$$

Q(0) Learning and Deep Q-Learning

A generalization of Watkins' algorithm [13, 26, 10]

Model Free Error Representation:

$\mathcal{E}^\theta(X_k, U_k) = -Q^\theta(X_k, U_k) + c(X_k, U_k) + \underline{Q}^\theta(X_{k+1})$

Goal as in the fixed-policy setting: Find roots of

$$\bar{f}(\theta) = \mathsf{E}_\infty[\zeta^\theta \mathcal{E}^\theta(X, U)] = \lim_{T\to\infty} \frac{1}{T} \sum_{k=0}^{T-1} \zeta_k^\theta \mathcal{E}^\theta(X_k, U_k)$$

Eligibility vector: $\zeta_k^\theta = \nabla_\theta Q^\theta(X_k, U_k)$ *Q(0)-learning*

Q(0) Learning and Deep Q-Learning

A generalization of Watkins' algorithm [13, 26, 10]

Model Free Error Representation:

$\mathcal{E}^\theta(X_k, U_k) = -Q^\theta(X_k, U_k) + c(X_k, U_k) + \underline{Q}^\theta(X_{k+1})$

Goal as in the fixed-policy setting: Find roots of

$$\bar{f}(\theta) = \mathsf{E}_\infty[\zeta^\theta \mathcal{E}^\theta(X, U)] = \lim_{T\to\infty} \frac{1}{T} \sum_{k=0}^{T-1} \zeta_k^\theta \mathcal{E}^\theta(X_k, U_k)$$

Eligibility vector: $\zeta_k^\theta = \nabla_\theta Q^\theta(X_k, U_k)$

Design principle:

Step 1: consider an ODE: $\frac{d}{dt}\theta_t = -G_t \bar{f}(\theta_t)$ (matrix gain part of design)

Step 2: translate to a discrete time algorithm based on measurements.

Q(0) Learning and Deep Q-Learning

A generalization of Watkins' algorithm [13, 26, 10]

Model Free Error Representation:

$\mathcal{E}^\theta(X_k, U_k) = -Q^\theta(X_k, U_k) + c(X_k, U_k) + \underline{Q}^\theta(X_{k+1})$

Goal as in the fixed-policy setting: Find roots of $\bar{f}(\theta^*) = 0$ *Why?*

$$\bar{f}(\theta) = \mathsf{E}_\infty[\zeta^\theta \mathcal{E}^\theta(X, U)] = \lim_{T\to\infty} \frac{1}{T} \sum_{k=0}^{T-1} \zeta_k^\theta \mathcal{E}^\theta(X_k, U_k)$$

Eligibility vector: $\zeta_k^\theta = \nabla_\theta Q^\theta(X_k, U_k)$

Design principle:
Step 1: consider an ODE: $\frac{d}{dt}\theta_t = -G_t \bar{f}(\theta_t)$ (matrix gain part of design)
Step 2: translate to a discrete time algorithm based on measurements.

Q(0) Learning and Deep Q-Learning

$$\bar{f}(\theta) = \lim \frac{1}{T} \sum_{k=0}^{T-1} \zeta_k^\theta \mathcal{E}^\theta(X_k, U_k) \qquad \bar{f}(\theta^*) = 0$$ *Why?*

Troubles with Q: Slow! Does a root exist? Does it have significance?

Q(0) Learning and Deep Q-Learning

$\bar{f}(\theta) = \lim \frac{1}{T} \sum_{k=0}^{T-1} \zeta_k^\theta \mathcal{E}^\theta(X_k, U_k)$ $\qquad \bar{f}(\theta^*) = 0$ *Why?*

Troubles with Q: Slow! Does a root exist? Does it have significance?

Batch algorithms to the rescue? [28, 29, 30, 31]

DQN $\theta_{n+1} = \arg\min_\theta \Big\{ \mathcal{E}_n(\theta) + \frac{1}{\alpha_{n+1}} \|\theta - \theta_n\|^2 \Big\}$

$$\mathcal{E}_n(\theta) = \frac{1}{r_n} \sum_{k=T_n}^{T_{n+1}-1} \big[-Q^\theta(X_k, U_k) + c(X_k, U_k) + \underline{Q}^{\theta_n}(X_{k+1}) \big]^2$$

Q(0) Learning and Deep Q-Learning

$\bar{f}(\theta) = \lim \frac{1}{T} \sum_{k=0}^{T-1} \zeta_k^\theta \mathcal{E}^\theta(X_k, U_k)$ $\qquad$ $\bar{f}(\theta^*) = 0$ $\qquad$ *Why?*

Troubles with Q: Slow! Does a root exist? Does it have significance?

Batch algorithms to the rescue? [28, 29, 30, 31]

DQN $\quad \theta_{n+1} = \arg\min_\theta \Big\{ \mathcal{E}_n(\theta) + \frac{1}{\alpha_{n+1}} \|\theta - \theta_n\|^2 \Big\}$

$$\mathcal{E}_n(\theta) = \frac{1}{r_n} \sum_{k=T_n}^{T_{n+1}-1} \big[-Q^\theta(X_k, U_k) + c(X_k, U_k) + \underline{Q}^{\theta_n}(X_{k+1}) \big]^2$$

With a linear parameterization, this is a quadratic program!

Q(0) Learning and Deep Q-Learning

$\bar{f}(\theta) = \lim \frac{1}{T} \sum_{k=0}^{T-1} \zeta_k^\theta \mathcal{E}^\theta(X_k, U_k)$ $\qquad \bar{f}(\theta^*) = 0$ $\quad$ *Why?*

Troubles with Q: Slow! Does a root exist? Does it have significance?

Batch algorithms to the rescue? [28, 29, 30, 31]

DQN $\quad \theta_{n+1} = \arg\min_\theta \Big\{ \mathcal{E}_n(\theta) + \frac{1}{\alpha_{n+1}} \|\theta - \theta_n\|^2 \Big\}$

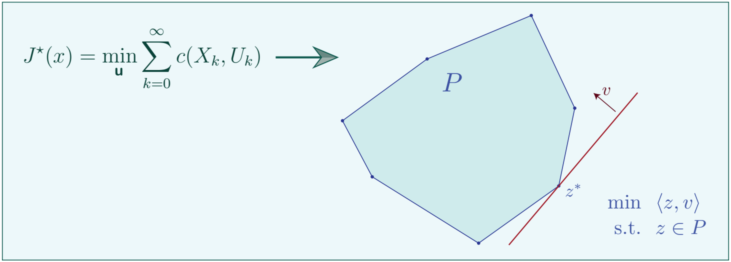
$$\mathcal{E}_n(\theta) = \frac{1}{r_n} \sum_{k=T_n}^{T_{n+1}-1} \big[-Q^\theta(X_k, U_k) + c(X_k, U_k) + \underline{Q}^{\theta_n}(X_{k+1}) \big]^2$$

With a linear parameterization, this is a quadratic program!

Sadly,

ODE approximation for DQN $\equiv$ Q(0) Learning

Even for neural network function approximation [M&M, 2020]



DP $\Rightarrow$ LP

Inverse Dynamic Programming

What is a good approximation? $\mathcal{E}(x) \stackrel{\text{def}}{=} -J(x) + \min_u [c(x,u) + J(\mathrm{F}(x,u))]$

Inverse Dynamic Programming

What is a good approximation? $\mathcal{E}(x) \stackrel{\text{def}}{=} -J(x) + \min_u [c(x,u) + J(\mathrm{F}(x,u))]$

For any J, you have solved a DP equation:

$$J(x) = \min_u [c_J(x,u) + J(\mathrm{F}(x,u))]$$

$$c_J(x,u) \stackrel{\text{def}}{=} c(x,u) - \mathcal{E}(x)$$

optimal policy ϕ^J

Let J^{ϕ^J} denote *the value function under the policy* ϕ^J

Inverse Dynamic Programming

What is a good approximation? $\mathcal{E}(x) \stackrel{\text{def}}{=} -J(x) + \min_u [c(x,u) + J(\mathrm{F}(x,u))]$

For any J, you have solved a DP equation:

$$J(x) = \min_u [c_J(x,u) + J(\mathrm{F}(x,u))]$$

$$c_J(x,u) \stackrel{\text{def}}{=} c(x,u) - \mathcal{E}(x)$$

optimal policy ϕ^J

Let J^{ϕ^J} denote *the value function under the policy* ϕ^J

Proposition 3.7

Assume $\mathcal{E}(x) \geq -\varrho c(x,u)$, all x, u and minor assumptions on J

Then, $J^\star(x) \leq J^{\phi^J}(x) \leq (1+\varrho) J^\star(x)$

Inverse Dynamic Programming

What is a good approximation? $\mathcal{E}(x) \stackrel{\text{def}}{=} -J(x) + \min_u [c(x,u) + J(\mathrm{F}(x,u))]$

For any J, you have solved a DP equation:

$$J(x) = \min_u [c_J(x,u) + J(\mathrm{F}(x,u))]$$

$$c_J(x,u) \stackrel{\text{def}}{=} c(x,u) - \mathcal{E}(x)$$

optimal policy ϕ^J

Let J^{ϕ^J} denote *the value function under the policy* ϕ^J

Proposition 3.7

Assume $\mathcal{E}(x) \geq -\varrho c(x,u)$, all x, u and minor assumptions on J

Then, $J^\star(x) \leq J^{\phi^J}(x) \leq (1+\varrho) J^\star(x)$

We have our gold standard

and our first LP constraint

Every DP is an LP

Every control student knows this, starting with [Manne, 1960] [44, 45, 46]

Proposition: [Subject to mild assumptions]
$J^\star$ solves the following LP:

$$\begin{aligned} \max_J \quad & \langle \mu, J \rangle \\ \text{s.t.} \quad & J(x) \le c(x,u) + J(\mathrm{F}(x,u))\,, \qquad x \in \mathsf{X}\,,\ u \in \mathsf{U}(x) \\ & J \text{ is continuous, and } J(x^e) = 0. \end{aligned}$$

μ a probability measure on X (given)

- Applications to ADP in the thesis of de Farias (with BVR) [47, 48], and Mengdi Wang's survey on Monday, August 31
- One way to derive the SDP representation of LQR [Boyd et al]
- Applications in deterministic control every day

Every DP is an LP

Every control student knows this, starting with [Manne, 1960] [44, 45, 46]

Proposition 3.9 [Subject to mild assumptions]
The pair $(J^\star, Q^\star)$ solve the following LP:

$$\begin{aligned} \max_{J,Q} \quad & \langle \mu, J \rangle \\ \text{s.t.} \quad & Q(x,u) \leq c(x,u) + J(\mathrm{F}(x,u)) \\ & Q(x,u) \geq J(x)\,, \qquad\qquad x \in \mathsf{X}\,,\ u \in \mathsf{U}(x) \\ & J \text{ is continuous, and } J(x^e) = 0. \end{aligned}$$

μ a probability measure on X (given)

Every DP is an LP

Every control student knows this, starting with [Manne, 1960] [44, 45, 46]

Proposition 3.9 [Subject to mild assumptions]
The pair $(J^\star, Q^\star)$ solve the following LP:

$$\begin{aligned} \max_{J,Q} \quad & \langle \mu, J \rangle \\ \text{s.t.} \quad & Q(x,u) \le c(x,u) + J(\mathrm{F}(x,u)) \\ & Q(x,u) \ge J(x)\,, \qquad\qquad x \in \mathsf{X}\,,\ u \in \mathsf{U}(x) \\ & J \text{ is continuous, and } J(x^e) = 0. \end{aligned}$$

μ a probability measure on X (given)

Over-parameterization for RL more recent.

Motivation: $Q(X_k, U_k) \le c(X_k, U_k) + J(X_{k+1})$ (observed)

Every DP is an LP

Explanation

Show that $J(x) \leq J^\star(x)$ for any feasible J, and all x

For any input sequence,
$J(X_k) \leq c(X_k, U_k) + J(X_{k+1})$

$$\implies J(X_0) \leq \sum_{k=0}^{T-1} c(X_k, U_k) + J(X_T)$$

$$\begin{aligned} \max_{J} \quad & \langle \mu, J \rangle \\ \text{s.t.} \quad & J(x) \leq c(x,u) + J(\mathrm{F}(x,u)) \\ & J \text{ is continuous, and } J(x^e) = 0. \end{aligned}$$

Every DP is an LP

Explanation

Show that $J(x) \leq J^\star(x)$ for any feasible J, and all x

For any input sequence,
$J(X_k) \leq c(X_k, U_k) + J(X_{k+1})$

$$\Longrightarrow J(X_0) \leq \sum_{k=0}^{T-1} c(X_k, U_k) + J(X_T)$$

$$\begin{aligned} \max_J \quad & \langle \mu, J \rangle \\ \text{s.t.} \quad & J(x) \leq c(x,u) + J(\mathrm{F}(x,u)) \\ & J \text{ is continuous, and } J(x^e) = 0. \end{aligned}$$

$J(X_T) \to 0$ for policies of interest, so

$$J(x) \leq \sum_{k=0}^{\infty} c(X_k, U_k)\,, \qquad X_0 = x$$

Every DP is an LP

Explanation

Show that $J(x) \leq J^\star(x)$ for any feasible J, and all x

$$\max_J \ \langle \mu, J \rangle$$
$$\text{s.t.} \quad J(x) \leq c(x,u) + J(\mathrm{F}(x,u))$$
$$J \text{ is continuous, and } J(x^e) = 0.$$

For any input sequence,
$J(X_k) \leq c(X_k, U_k) + J(X_{k+1})$

$$\Longrightarrow J(X_0) \leq \sum_{k=0}^{T-1} c(X_k, U_k) + J(X_T)$$

$J(X_T) \to 0$ for policies of interest, so

$$J(x) \leq \sum_{k=0}^{\infty} c(X_k, U_k)\,, \qquad X_0 = x$$

Take the infimum over all $\boldsymbol{U} \implies$ QED

Every DP is a QP

Proposition: [Subject to mild assumptions]
The pair $(J^\star, Q^\star)$ solve the following QP:

$$\min_{J,Q} \quad -\langle \mu, J\rangle + \kappa\langle \nu, \mathcal{E}^2\rangle$$

$$\text{s.t.} \quad 0 \leq \mathcal{E}(x,u) \overset{\text{def}}{=} -Q(x,u) + c(x,u) + J(\mathrm{F}(x,u))$$

$$Q(x,u) \geq J(x)\,, \qquad x \in \mathsf{X}\,,\ u \in \mathsf{U}(x)$$

J is continuous, and $J(x^e) = 0$.

ν a probability measure on $\mathsf{X} \times \mathsf{U}$

Every DP is a QP

Proposition: [Subject to mild assumptions]
The pair $(J^\star, Q^\star)$ solve the following QP:

$$\min_{J,Q} \quad -\langle \mu, J\rangle + \kappa\langle \nu, \mathcal{E}^2\rangle$$

$$\text{s.t.} \quad 0 \le \mathcal{E}(x,u) \stackrel{\text{def}}{=} -Q(x,u) + c(x,u) + J(\mathrm{F}(x,u))$$

$$Q(x,u) \ge J(x)\,, \qquad x \in \mathsf{X}\,,\ u \in \mathsf{U}(x)$$

J is continuous, and $J(x^e) = 0$.

ν a probability measure on $\mathsf{X} \times \mathsf{U}$

The objective and constraints can be observed, without a model
$\Longrightarrow$ Long list of possible RL approximations

Convex Q-Learning

Every DP is a QP $\Longrightarrow$ Convex Q Learning

$$
\begin{aligned}
\min_{\theta} \quad & -\langle \mu, J^\theta \rangle + \kappa \langle \nu, \mathcal{E}^2(\theta) \rangle \\
\text{s.t.} \quad & 0 \le -Q^\theta(X_k, U_k) + c(X_k, U_k) + J^\theta(X_{k+1}) \\
& Q^\theta(x, u) \ge J^\theta(x) \qquad \Longleftarrow \textit{Enforce through function architecture}
\end{aligned}
$$

Every DP is a QP $\Longrightarrow$ Convex Q Learning

$$\begin{aligned} \min_{\theta} \quad & -\langle \mu, J^\theta \rangle + \kappa \langle \nu, \mathcal{E}^2(\theta) \rangle \\ \text{s.t.} \quad & 0 \le -Q^\theta(X_k, U_k) + c(X_k, U_k) + J^\theta(X_{k+1}) \qquad \Longrightarrow \qquad z_n(\theta) \ge 0 \end{aligned}$$

$$z_n(\theta) = \frac{1}{r_n} \sum_{k=T_n}^{T_{n+1}-1} \big[-Q^\theta(X_k, U_k) + c(X_k, U_k) + J^\theta(X_{k+1}) \big] \zeta_k^+$$

ζ_k^+ : vector with non-negative entries

Every DP is a QP $\Longrightarrow$ Convex Q Learning

$$\min_{\theta} \quad -\langle \mu, J^\theta \rangle + \kappa \langle \nu, \mathcal{E}^2(\theta) \rangle$$

$$\text{s.t.} \quad 0 \le -Q^\theta(X_k, U_k) + c(X_k, U_k) + J^\theta(X_{k+1}) \quad \Longrightarrow \quad z_n(\theta) \ge 0$$

$$z_n(\theta) = \frac{1}{r_n} \sum_{k=T_n}^{T_{n+1}-1} \big[-Q^\theta(X_k, U_k) + c(X_k, U_k) + J^\theta(X_{k+1}) \big] \zeta_k^+$$

$$\bar{\mathcal{E}}_n^2(\theta) = \frac{1}{r_n} \sum_{k=T_n}^{T_{n+1}-1} \big[-Q^\theta(X_k, U_k) + c(X_k, U_k) + J^\theta(X_{k+1}) \big]^2$$

Every DP is a QP $\Longrightarrow$ Convex Q Learning

$$
\begin{aligned}
\min_{\theta} \quad & -\langle \mu, J^\theta \rangle + \kappa \langle \nu, \mathcal{E}^2(\theta) \rangle \\
\text{s.t.} \quad & 0 \le -Q^\theta(X_k, U_k) + c(X_k, U_k) + J^\theta(X_{k+1}) \quad \Longrightarrow \quad z_n(\theta) \ge 0
\end{aligned}
$$

$$
z_n(\theta) = \frac{1}{r_n} \sum_{k=T_n}^{T_{n+1}-1} [-Q^\theta(X_k, U_k) + c(X_k, U_k) + J^\theta(X_{k+1})] \zeta_k^+
$$

$$
\bar{\mathcal{E}}_n^2(\theta) = \frac{1}{r_n} \sum_{k=T_n}^{T_{n+1}-1} [-Q^\theta(X_k, U_k) + c(X_k, U_k) + J^\theta(X_{k+1})]^2
$$

Convex Q Version 1.0

$$
\theta_{n+1} = \operatorname*{arg\,min}_{\theta \in \Theta} \Big\{ -\langle \mu, J^\theta \rangle + \kappa \bar{\mathcal{E}}_n^2(\theta) - \lambda_n^{\intercal} z_n(\theta) + \frac{1}{\alpha_{n+1}} \tfrac{1}{2} \|\theta - \theta_n\|^2 \Big\}
$$

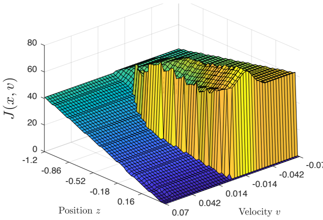
$$
\lambda_{n+1} = \big[\lambda_n - \alpha_{n+1} z_n(\theta_n) \big]_+
$$

Convex Q Learning—Does it work?

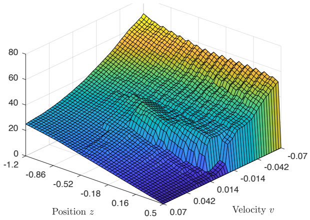
$$\theta_{n+1} = \mathop{\arg\min}_{\theta\in\Theta}\Big\{-\langle \mu, J^\theta\rangle + \kappa\bar{\mathcal{E}}_n^2(\theta) - \lambda_n^T z_n(\theta) + \frac{1}{\alpha_{n+1}}\tfrac{1}{2}\|\theta-\theta_n\|^2\Big\}$$

$$\lambda_{n+1} = \big[\lambda_n - \alpha_{n+1} z_n(\theta_n)\big]_+$$

It is only 4 weeks old! Who knows what Version 1.1 will look like.



Value function obtained from VIA



Value function approximation from convex Q

MountainCar in early August

Convex Q Learning—Does it work?

$$\theta_{n+1} = \mathop{\arg\min}_{\theta\in\Theta}\Big\{-\langle \mu, J^\theta\rangle + \kappa\bar{\mathcal{E}}_n^2(\theta) - \lambda_n^{\intercal} z_n(\theta) + \frac{1}{\alpha_{n+1}}\tfrac{1}{2}\|\theta-\theta_n\|^2\Big\}$$

$$\lambda_{n+1} = \big[\lambda_n - \alpha_{n+1} z_n(\theta_n)\big]_+$$

Lessons learned from initial testing:

1. Advantage function: $A^\theta = Q^\theta - J^\theta$, with Θ chosen so
   $A^\theta(x,u) \ge 0$ all $x, u, \theta \in \Theta$
   Seems necessary for success

Convex Q Learning—Does it work?

$$\begin{aligned}\theta_{n+1} &= \operatorname*{arg\,min}_{\theta\in\Theta}\Big\{-\langle\mu, J^\theta\rangle + \kappa\bar{\mathcal{E}}_n^2(\theta) - \lambda_n^{\intercal} z_n(\theta) + \frac{1}{\alpha_{n+1}}\tfrac{1}{2}\|\theta-\theta_n\|^2\Big\}\\ \lambda_{n+1} &= \big[\lambda_n - \alpha_{n+1} z_n(\theta_n)\big]_+\end{aligned}$$

Lessons learned from initial testing:

1. Advantage function: $A^\theta = Q^\theta - J^\theta$, with Θ chosen so
   $A^\theta(x,u) \ge 0$ all $x, u, \theta \in \Theta$
2. There may be sensitivity here: $\langle \mu, J\rangle$

Convex Q Learning—Does it work?

$$\theta_{n+1} = \operatorname*{arg\,min}_{\theta \in \Theta} \Big\{ -\langle \mu, J^\theta \rangle + \kappa \bar{\mathcal{E}}_n^2(\theta) - \lambda_n^{\mathsf{T}} z_n(\theta) + \frac{1}{\alpha_{n+1}} \tfrac{1}{2} \|\theta - \theta_n\|^2 \Big\}$$

$$\lambda_{n+1} = \big[\lambda_n - \alpha_{n+1} z_n(\theta_n)\big]_+$$

Lessons learned from initial testing:

1. Advantage function: $A^\theta = Q^\theta - J^\theta$, with Θ chosen so
   $$A^\theta(x,u) \ge 0 \text{ all } x, u, \theta \in \Theta$$
2. There may be sensitivity here: $\langle \mu, J \rangle$
3. Many problems on `openai.com` are difficult because of fast sampling:

$$X_{k+1} \approx X_k \Longrightarrow -Q^\theta(X_k, U_k) + c(X_k, U_k) + J^\theta(X_{k+1}) \approx -A^\theta(X_k, U_k) + c(X_k, U_k)$$

Convex Q Learning—Does it work?

$$\theta_{n+1} = \mathop{\arg\min}_{\theta\in\Theta}\Big\{-\langle\mu, J^\theta\rangle + \kappa\bar{\mathcal{E}}_n^2(\theta) - \lambda_n^T z_n(\theta) + \frac{1}{\alpha_{n+1}}\tfrac{1}{2}\|\theta-\theta_n\|^2\Big\}$$

$$\lambda_{n+1} = \big[\lambda_n - \alpha_{n+1} z_n(\theta_n)\big]_+$$

Lessons learned from initial testing:

1. Advantage function: $A^\theta = Q^\theta - J^\theta$, with Θ chosen so
   $A^\theta(x,u) \ge 0$ all $x, u, \theta \in \Theta$
2. There may be sensitivity here: $\langle\mu, J\rangle$
3. Many problems on `openai.com` are difficult because of fast sampling:

$$X_{k+1} \approx X_k \Longrightarrow -Q^\theta(X_k,U_k) + c(X_k,U_k) + J^\theta(X_{k+1}) \approx -A^\theta(X_k,U_k) + c(X_k,U_k)$$

Revisit Convex Q in continuous time [M&M 09]

Conclusions

The LP and QP characterization of DP equations gives rise to RL algorithms that are provably convergent, and for which we know what problem we are actually solving!

Conclusions

The LP and QP characterization of DP equations gives rise to RL algorithms that are provably convergent, and for which we know what problem we are actually solving!

- Extensions to stochastic control – *not a big deal*
- Much more work is required to develop these algorithms for particular applications, and to improve efficiency

Conclusions

The LP and QP characterization of DP equations gives rise to RL algorithms that are provably convergent, and for which we know what problem we are actually solving!

- Extensions to stochastic control – *not a big deal*
- Much more work is required to develop these algorithms for particular applications, and to improve efficiency
- Extensions to Convex Policy Gradient: Manne's LP suggests we parametrize desired occupancy probabilities, and not the policy

Conclusions

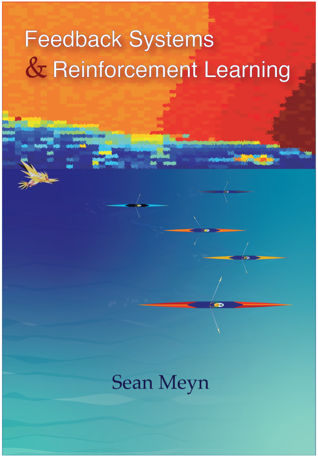
The LP and QP characterization of DP equations gives rise to RL algorithms that are provably convergent, and for which we know what problem we are actually solving!

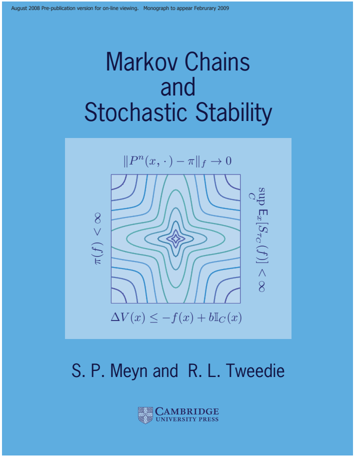
- Extensions to stochastic control – *not a big deal*
- Much more work is required to develop these algorithms for particular applications, and to improve efficiency
- Extensions to Convex Policy Gradient: Manne's LP suggests we parametrize desired occupancy probabilities, and not the policy
- More today:
  - Explain ODE method and "ODE approximation for DQN..."

Conclusions

The LP and QP characterization of DP equations gives rise to RL algorithms that are provably convergent, and for which we know what problem we are actually solving!

- Extensions to stochastic control – *not a big deal*
- Much more work is required to develop these algorithms for particular applications, and to improve efficiency
- Extensions to Convex Policy Gradient: Manne's LP suggests we parametrize desired occupancy probabilities, and not the policy
- More today:
  - Explain ODE method and "ODE approximation for DQN..."
  - QSA
  - qSGD
  - qPG


Feedback Systems
& Reinforcement Learning
Sean Meyn


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 R. M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, USA, 2008 (recent edition on-line).

[2] K. J. Åström and K. Furuta. Swinging up a pendulum by energy control. *Automatica*, 36(2):287 – 295, 2000.

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

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

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

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

[7] B. Wittenmark. Stochastic adaptive control methods: a survey. *International Journal of Control*, 21(5):705–730, 1975.

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

Control Background II

[9] 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

[10] 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.

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

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

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

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

[15] T. Jaakola, M. Jordan, and S. Singh. On the convergence of stochastic iterative dynamic programming algorithms. *Neural Computation*, 6:1185–1201, 1994.

[16] 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.

[17] 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.

RL Background II

[18] 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.

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

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

[21] 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.

[22] 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.

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

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

RL Background III

[25] 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.

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

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

DQN:

[28] 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.

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

[30] 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 IV

[31] 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.

Actor Critic / Policy Gradient

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

[33] 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.

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

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

[36] 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.

RL Background V

[37] 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.

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

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

[40] 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.

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

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

RL Background VI

[43] 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:

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

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

[46] 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.

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

[48] 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.

RL Background VII

[49] 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.

[50] P. G. Mehta and S. P. Meyn. *Convex Q-learning, part 1: Deterministic optimal control. ArXiv e-prints:2008.03559*, 2020.

Gator Nation:

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

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

[53] A. M. Devraj and S. P. Meyn. Q-learning with Uniformly Bounded Variance: Large Discounting is Not a Barrier to Fast Learning. *arXiv e-prints 2002.10301, and to appear AISTATS*, Feb. 2020.

[54] 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.

Stochastic Miscellanea I

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

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

[57] 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.

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

Stochastic Approximation I

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

[60] 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.

[61] 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.

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

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

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

[65] 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 II

[66] 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*.

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

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

[69] 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.

[70] S. Chen, A. M. Devraj, A. Bušić, and S. Meyn. Explicit Mean-Square Error Bounds for Monte-Carlo and Linear Stochastic Approximation. *arXiv e-prints*, 2002.02584, Feb. 2020.

[71] 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

[72] 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.

[73] 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.

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

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

QSA and Extremum Seeking Control I

[76] S. Chen, A. Bernstein, A. Devraj, and S. Meyn. Accelerating optimization and reinforcement learning with quasi-stochastic approximation. *arXiv:In preparation*, 2020.

[77] B. Lapeybe, G. Pages, and K. Sab. Sequences with low discrepancy generalisation and application to Robbins-Monro algorithm. *Statistics*, 21(2):251–272, 1990.

[78] S. Laruelle and G. Pagès. Stochastic approximation with averaging innovation applied to finance. *Monte Carlo Methods and Applications*, 18(1):1–51, 2012.

[79] S. Shirodkar and S. Meyn. Quasi stochastic approximation. In *Proc. of the 2011 American Control Conference (ACC)*, pages 2429–2435, July 2011.

[80] A. Bernstein, Y. Chen, M. Colombino, E. Dall'Anese, P. Mehta, and S. Meyn. Optimal rate of convergence for quasi-stochastic approximation. *arXiv:1903.07228*, 2019.

[81] A. Bernstein, Y. Chen, M. Colombino, E. Dall'Anese, P. Mehta, and S. Meyn. Quasi-stochastic approximation and off-policy reinforcement learning. In *Proc. of the IEEE Conf. on Dec. and Control*, pages 5244–5251, Mar 2019.

[82] Y. Chen, A. Bernstein, A. Devraj, and S. Meyn. Model-Free Primal-Dual Methods for Network Optimization with Application to Real-Time Optimal Power Flow. In *Proc. of the American Control Conf.*, pages 3140–3147, Sept. 2019.

QSA and Extremum Seeking Control II

[83] S. Bhatnagar and V. S. Borkar. Multiscale chaotic spsa and smoothed functional algorithms for simulation optimization. *Simulation*, 79(10):568–580, 2003.

[84] S. Bhatnagar, M. C. Fu, S. I. Marcus, and I.-J. Wang. Two-timescale simultaneous perturbation stochastic approximation using deterministic perturbation sequences. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 13(2):180–209, 2003.

[85] M. Le Blanc. Sur l'electrification des chemins de fer au moyen de courants alternatifs de frequence elevee [On the electrification of railways by means of alternating currents of high frequency]. *Revue Generale de l'Electricite*, 12(8):275–277, 1922.

[86] Y. Tan, W. H. Moase, C. Manzie, D. Nešić, and I. M. Y. Mareels. Extremum seeking from 1922 to 2010. In *Proceedings of the 29th Chinese Control Conference*, pages 14–26, July 2010.

[87] P. F. Blackman. Extremum-seeking regulators. In *An Exposition of Adaptive Control*. Macmillan, 1962.

[88] J. Sternby. Adaptive control of extremum systems. In H. Unbehauen, editor, *Methods and Applications in Adaptive Control*, pages 151–160, Berlin, Heidelberg, 1980. Springer Berlin Heidelberg.

QSA and Extremum Seeking Control III

[89] J. Sternby. Extremum control systems–an area for adaptive control? In *Joint Automatic Control Conference*, number 17, page 8, 1980.

[90] K. B. Ariyur and M. Krstić. *Real Time Optimization by Extremum Seeking Control*. John Wiley & Sons, Inc., New York, NY, USA, 2003.

[91] M. Krstić and H.-H. Wang. Stability of extremum seeking feedback for general nonlinear dynamic systems. *Automatica*, 36(4):595 – 601, 2000.

[92] S. Liu and M. Krstic. Introduction to extremum seeking. In *Stochastic Averaging and Stochastic Extremum Seeking*, Communications and Control Engineering. Springer, London, 2012.

[93] O. Trollberg and E. W. Jacobsen. On the convergence rate of extremum seeking control. In *European Control Conference (ECC)*, pages 2115–2120. 2014.

Selected Applications I

[94] N. S. Raman, A. M. Devraj, P. Barooah, and S. P. Meyn. *Reinforcement learning for control of building HVAC systems.* In *American Control Conference*, July 2020.

[95] K. Mason and S. Grijalva. A review of reinforcement learning for autonomous building energy management. *arXiv.org*, 2019. arXiv:1903.05196.

News from Andrey@NREL:

[96] A. Bernstein and E. Dall'Anese. Real-time feedback-based optimization of distribution grids: A unified approach. *IEEE Transactions on Control of Network Systems*, 6(3):1197–1209, 2019.

[97] A. Bernstein, E. Dall'Anese, and A. Simonetto. Online primal-dual methods with measurement feedback for time-varying convex optimization. *IEEE Transactions on Signal Processing*, 67(8):1978–1991, 2019.

[98] Y. Chen, A. Bernstein, A. Devraj, and S. Meyn. Model-free primal-dual methods for network optimization with application to real-time optimal power flow. In *2020 American Control Conference (ACC)*, pages 3140–3147, 2020.