Fast rates for online learning in Linearly Solvable Markov Decision Processes

Gergely Neu & Vicenç Gómez

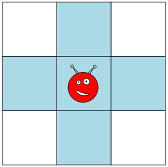
Universitat Pompeu Fabra
Barcelona, Spain

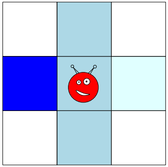
Linearly Solvable Markov Decision Processes

"Offline" version [Todorov, 2010, Kappen, 2005]

Control a sequence of states $X_1, X_2, \ldots$ trying to

- minimize a state cost $c : X \mapsto [0, 1]$
- not deviate too much from the passive dynamics $P(X'|X)$





Linearly Solvable Markov Decision Processes

"Offline" version [Todorov, 2010, Kappen, 2005]

Repeat for $t = 1, 2, \ldots$:

- LEARNER
  - observes state X_t and picks next-state distribution $Q_t(\cdot|X_t)$
  - suffers loss

$$\ell(X_t, Q_t) = c(X_t) + \sum_x Q_t(x|X_t) \log \frac{Q_t(x|X_t)}{P(x|X_t)}$$

- ENVIRONMENT generates next state $X_{t+1} \sim Q_t(\cdot|X_t)$.

Linearly Solvable Markov Decision Processes

"Offline" version [Todorov, 2010, Kappen, 2005]

Repeat for $t = 1, 2, \ldots$:

- LEARNER
  - observes state X_t and picks next-state distribution $Q_t(\cdot|X_t)$
  - suffers loss

$$\ell(X_t, Q_t) = c(X_t) + \sum_x Q_t(x|X_t) \log \frac{Q_t(x|X_t)}{P(x|X_t)}$$

- ENVIRONMENT generates next state $X_{t+1} \sim Q_t(\cdot|X_t)$.

GOAL: minimize average cost-per stage

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \ell(X_t, Q_t) \to \min$$

Linearly Solvable Markov Decision Processes

"Offline" version [Todorov, 2010, Kappen, 2005]

Optimal policy given by

$$Q(x'|x) = \frac{P(x'|x)z(x')}{\sum_y P(y|x)z(y)}.$$

where z is the solution to the eigenvalue problem

$$e^{-\lambda} z = \operatorname{diag}\left(e^{-c(x)}\right) Pz.$$

Linearly Solvable Markov Decision Processes

"Offline" version [Todorov, 2010, Kappen, 2005]

"Linearly solvable"

Optimal policy given by

$$Q(x'|x) = \frac{P(x'|x)z(x')}{\sum_y P(y|x)z(y)}.$$

where z is the solution to the eigenvalue problem

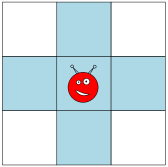
$$e^{-\lambda} z = \text{diag}\left(e^{-c(x)}\right) Pz.$$

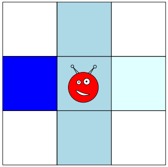
Linearly Solvable Markov Decision Processes

"Offline" version [Todorov, 2010, Kappen, 2005]

Control a sequence of states $X_1, X_2, \ldots$ trying to

- minimize a state cost $c : X \mapsto [0, 1]$
- not deviate too much from the passive dynamics $P(X'|X)$



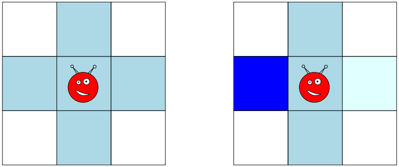


This work: Online learning in LMDPs

First studied by Guan, Raginsky, and Willett [2014]

Control a sequence of states $X_1, X_2, \ldots$ trying to

- minimize a sequence of state costs $c_t : X \mapsto [0, 1]$
- not deviate too much from the passive dynamics $P(X'|X)$



Online learning in LMDPs

First studied by Guan, Raginsky, and Willett [2014]

Repeat for $t = 1, 2, \dots$:

- LEARNER
  - observes state X_t and picks next-state distribution $Q_t(\cdot|X_t)$
  - suffers loss

$$\ell(X_t, Q_t) = c_t(X_t) + \sum_x Q_t(x|X_t) \log \frac{Q_t(x|X_t)}{P(x|X_t)}$$

- ENVIRONMENT
  - generates next state $X_{t+1} \sim Q_t(\cdot|X_t)$,
  - picks state-cost function $c_t : X \mapsto [0, 1]$

Online learning in LMDPs

First studied by Guan, Raginsky, and Willett [2014]

Repeat for $t = 1, 2, \ldots$:

- LEARNER
  - observes state X_t and picks next-state distribution $Q_t(\cdot|X_t)$
  - suffers loss

$$\ell(X_t, Q_t) = c_t(X_t) + \sum_x Q_t(x|X_t) \log \frac{Q_t(x|X_t)}{P(x|X_t)}$$

- ENVIRONMENT
  - generates next state $X_{t+1} \sim Q_t(\cdot|X_t)$,
  - picks state-cost function $c_t : X \mapsto [0, 1]$

GOAL: minimize regret

$$R_T = \max_Q \sum_{t=1}^{T} \mathbb{E}\left[\ell_t(X_t, Q_t) - \ell_t(X_t, Q)\right]$$

Online learning in LMDPs

State of the art [Guan, Raginsky, and Willett, 2014]:

$$R_T = O\left(T^{3/4+\varepsilon}\right)$$

Online learning in LMDPs

State of the art [Guan, Raginsky, and Willett, 2014]:

$$R_T = O\left(T^{3/4+\varepsilon}\right)$$

Open problem: can this be improved to $O\left(\sqrt{T}\right)$?

Online learning in LMDPs

State of the art [Guan, Raginsky, and Willett, 2014]:

$$R_T = O\left(T^{3/4+\varepsilon}\right)$$

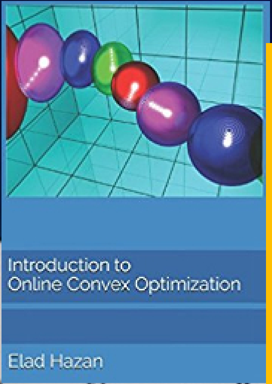
Open problem: can this be improved to $O\left(\sqrt{T}\right)$?

Our result: $R_T = O\left(\log^2 T\right)$

(same assumptions: bounded 1-step mixing time of passive dynamics)

The secret sauce

The secret sauce



Introduce idealized problem:

Repeat for $t = 1, 2, \dots$:

LEARNER:

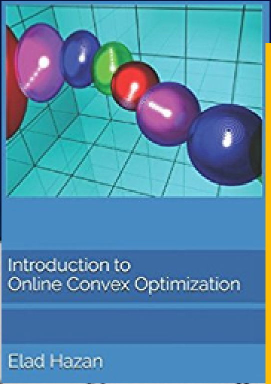
- picks stationary distribution $\pi_t \in \Delta(\mathcal{X}^2)$
- suffers loss $\widetilde{\ell}_t(\pi_t) = \langle \pi_t, c_t \rangle + R(\pi_t)$, where

$$R(\pi) = \sum_{x,x'} \pi(x,x') \log \frac{\pi(x,x')}{P(x'|x) \sum_y \pi(x,y)}$$

ENVIRONMENT

- picks state-cost function $c_t : X \mapsto [0,1]$

The secret sauce



Introduce idealized problem:

Repeat for $t = 1, 2, \ldots$:

LEARNER:

- picks stationary distribution $\pi_t \in \Delta(\mathcal{X}^2)$
- suffers loss $\widetilde{\ell}_t(\pi_t) = \langle \pi_t, c_t \rangle + R(\pi_t)$, where

$$R(\pi) = \sum_{x,x'} \pi(x, x') \log \frac{\pi(x, x')}{P(x'|x) \sum_y \pi(x, y)}$$

ENVIRONMENT

- picks state-cost function $c_t : X \mapsto [0, 1]$

$R(\pi)$: the conditional entropy of $(X', X) \sim \pi$

... a convex function of π!

The algorithm & the rest of the proof

Algorithm: Follow the Leader:

$$\pi_t = \arg\min_{\pi} \sum_{s=1}^{t-1} \widetilde{\ell}_s(\pi)$$

The algorithm & the rest of the proof

Algorithm: Follow the Leader:

$$\pi_t = \arg\min_{\pi} \sum_{s=1}^{t-1} \widetilde{\ell}_s(\pi)$$

Analysis:

- Show that policies change smoothly: $\|\pi_t - \pi_{t+1}\|_1 = O(1/t)$
- Bound idealized regret by $O(\log T)$ (FTL/BTL lemma)
- Gap between idealized and true regret $= O(\log^2 T)$
- + a bunch of technical tools taken from Guan, Raginsky, and Willett [2014]...

References I

P. Guan, M. Raginsky, and R. M. Willett. Online markov decision processes with kullback–leibler control cost. *Automatic Control, IEEE Transactions on*, 59(6):1423–1438, 2014.

H. J. Kappen. Linear theory for control of nonlinear stochastic systems. *Physical review letters*, 95(20):200201, 2005.

E. Todorov. Policy gradients in linearly-solvable mdps. In *NIPS-23*, pages 2298–2306. CURRAN, 2010.