Politex: Regret Bounds for Policy Iteration Using Expert Prediction

Yasin Abbasi-Yadkori [1], Peter L. Bartlett [2], Kush Bhatia [2], Nevena Lazić [3], Csaba Szepesvári [4], Gellért Weisz [4]

[1]Adobe, [2]Berkeley, [3]Google, [4]DeepMind

Setting and notation

- Markov decision process (MDP)
  observed states $x \in \mathcal{S}$,
  discrete actions $a \in \{1, \ldots, A\}$,
  unknown costs $c(x, a)$
  unknown transition dynamics $P(x_{t+1}|x_t, a_t)$
- Average cost of a policy $\pi(a|x)$: [1]

$$\lambda_\pi = \mathbf{E}\left[\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} c(x_t^\pi, a_t^\pi) \right]$$

A1 Unichain: MDP states form a single recurrent class under any policy.

A2 Uniform mixing: $\|(\nu' - \nu_\pi)H\|_1 \leq \exp(-1/\kappa)\|\nu' - \nu_\pi\|$, where $\nu_\pi(x, a)$ is the steady-state distribution of π, and $H_{(x,a),(x',a')} = P(x'|x, a)\pi(a'|x')$.

[1] $\{(x_t^\pi, a_t^\pi)\}_{t=1,2,\ldots}$ denotes the state-action sequence when following π

Policy iteration

Input: phase length $\tau > 0$, initial state x_0

Set $\widehat{Q}_0(x, a) = 0, \;\; \pi_0(a|x) = 1/A \;\; \forall x, a$
for $i := 0, 1, 2, \ldots,$ **do**
Policy evaluation:
Execute π_i for τ time steps and collect data.
Compute the action-value estimate $\widehat{Q}_i(x, a)$.

Policy improvement:

$$\pi_{i+1}(\cdot|x) = \operatorname*{argmin}_{u \in \Delta} \langle u, \widehat{Q}_i(x, \cdot) \rangle$$

end for

$$Q_\pi(x, a) = c(x, a) - \lambda_\pi + \mathbf{E}\left[\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} c(x_t^\pi, a_t^\pi) \middle| x_0 = x, a_0 = a \right]$$

$\widehat{Q}_i$ is an approximation of Q_{π_i} (e.g. linear or neural network)

Policy iteration using expert advice (POLITEX)

Input: phase length $\tau > 0$, initial state x_0

Set $\widehat{Q}_0(x, a) = 0, \;\; \pi_0(a|x) = 1/A \;\; \forall x, a$

for $i := 0, 1, 2, \ldots,$ **do**

Policy evaluation:

Execute π_i for τ time steps and collect data.

Compute the action-value estimate $\widehat{Q}_i(x, a)$.

Policy improvement:

$$\pi_{i+1}(\cdot|x) = \operatorname*{argmin}_{u \in \Delta} \langle u, \sum_{j=0}^{i} \widehat{Q}_j(x, \cdot) \rangle - \eta^{-1} \mathcal{H}(u)$$

$$\propto \exp\left(-\eta \sum_{j=0}^{i} \widehat{Q}_j(x, \cdot) \right)$$

end for

POLITEX regret (informal)

- For $\widehat{Q}_i$ estimated from τ transitions, we require

$$\widehat{Q}_i \in [b, b + Q_{\max}] \quad \text{and} \quad \|Q_{\pi_i} - \widehat{Q}_i\|_{\nu_{\pi_i}} = \varepsilon_0 + O(1/\sqrt{\tau}),$$

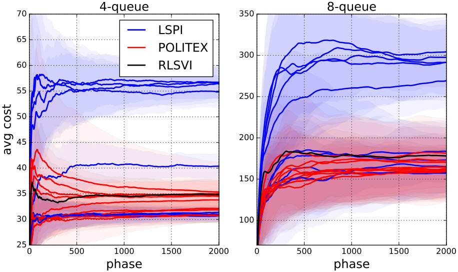
  where ε_0 is the approximation error. Satisfied e.g. by LSPE (Bertsekas & Ioffe, 1996) under a "feature excitation" assumption on the policies.
- Then the regret of POLITEX w.r.t a reference policy π^*, defined as $\mathfrak{R}_T = \sum_{t=1}^{T} c(x_t, a_t) - c(x_t^*, a_t^*)$, is of the order

$$\mathfrak{R}_T = \widetilde{O}(T^{3/4} + \varepsilon_0 T).$$

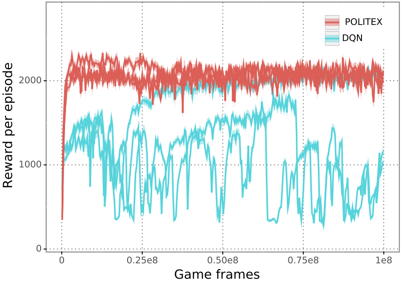
- Regret bound does not scale in the size of the underlying MDP.
- Unlike existing policy iteration results (for discounted MDPs), does not depend on the concentrability coefficient.
- Easy to implement - no confidence bounds required.

Experiments

Politex + LSPE on Queueing networks



Politex + neural nets on Ms Pacman



Related work

- E. Even-Dar, S. Kakade, and Y. Mansour, *Online MDPs.* Mathematics of Operations Research, 2009.

  MDP-E uses an experts algorithm in each state x with losses $Q(x, a)$. POLITEX is similar, but learns the action-value function from data.

- Y. Abbasi-Yadkori, N. Lazić, and Cs. Szepesvári, *Regret bounds for model-free linear quadratic control via reduction to expert prediction.* AISTATS 2019.

  Similar approach applied to the control of LQ systems.

- H. Yu and D. Bertsekas, *Convergence results for some temporal difference methods based on least squares.* IEEE Transactions on Automatic Control, 2009.

  Asymptotic convergence analysis of average-cost LSPE, here adapted to finite-sample analysis for learning Q functions.

- Degrave et al., *Quinoa.* NeurIPS DeepRL Workshop, 2018.
  Abdolmaleki et al, *Maximum a-posteriori policy optimization.* ICLR, 2018.

  Similar algorithms based on heuristics.