UNIVERSITY OF ALBERTA

Revisiting Peng's Q(λ): Good Old-Fashioned Algorithm for Modern Reinforcement Learning

Tadashi Kozuno[1*], Yunhao Tang[2*], Mark Rowland[3], Remi Munos[4], Steven Kapturowski[3], Will Dabney[3], Michal Valko[4], David Abel[3]

*: equal contribution

1: Univ of Alberta, 2: Columbia Univ, 3: DeepMind London, 4: DeepMind Paris

Value Iteration (VI)

Initialize a Q-function Q;

Initialize a behavior policy μ;

For $t = 1, \dots, T$ do;

Observe s_t and take $a_t \sim \mu(\cdot \mid a_t)$.

Observe s_{t+1} and receive r_t.

Store (s_t, a_t, s_{t+1}, r_t) in data buffer $\mathcal{D}$.

If $t \bmod T_{update} = 0$ then

$Q \leftarrow \mathrm{argmin}_f \mathbb{E}_{\mathcal{D}}\left[\left(f(s,a) - \hat{Q}_{VI}\right)^2\right]$.

$\mu \leftarrow \mathrm{NewBehaviorPolicy}(Q)$.

Value Iteration (VI)

Initialize a Q-function Q;

Initialize a behavior policy μ;

For $t = 1, \ldots, T$ do;

Observe s_t and take $a_t \sim \mu(\cdot \mid a_t)$.

Observe s_{t+1} and receive r_t.

Store (s_t, a_t, s_{t+1}, r_t) in data buffer $\mathcal{D}$.

If $t \bmod T_{update} = 0$ then

$Q \leftarrow \mathrm{argmin}_f \mathbb{E}_{\mathcal{D}} \left[\left(f(s,a) - \hat{Q}_{VI} \right)^2 \right]$.

$\mu \leftarrow \mathrm{NewBehaviorPolicy}(Q)$.

$$Q_{VI} \coloneqq r + \gamma \max_{a'} Q(s', a')$$

VI to λ Policy Iteration (λ-PI)

λ Policy Iteration:

$$Q_{\lambda PI} \coloneqq (1-\lambda)\sum_{N=1}^{\infty}\lambda^{N-1}\left(\sum_{n=1}^{N}\gamma^{n-1}r_{t+n}+\gamma^{N}\max_{a}Q(s_{t+N+1},a)\right)$$

$$= Q(s_t,a_t)+\sum_{N=0}^{\infty}\gamma^{N}\lambda^{N}\left(r_{t+N}+\gamma\max_{a}Q(s_{t+N+1},a)-Q(s_{t+N},a_{t+N})\right)$$

$(s_{t+n},a_{t+n})_{n=0,1,\ldots,}$ is obtained by following a greedy policy!

Importance Sampling λ-PI

Importance sampling λ-PI:

$$Q_{IS\,\lambda PI} \coloneqq Q(s_t, a_t) + \sum_{N=0}^{\infty} c_N \gamma^N \lambda^N \left(r_{t+N} + \gamma \max_a Q(s_{t+N+1}, a) - Q(s_{t+N}, a_{t+N}) \right)$$

where $c_0 \coloneqq 1$, and $c_N \coloneqq \prod_{n=0}^{N} \frac{\pi(a_{t+n}|s_{t+n})}{\mu(a_{t+n}|s_{t+n})}$.

Importance Sampling λ -PI

Importance sampling λ -PI:

$$Q_{IS\,\lambda PI} \coloneqq Q(s_t, a_t) + \sum_{N=0}^{\infty} c_N \gamma^N \lambda^N \left(r_{t+N} + \gamma \max_a Q(s_{t+N+1}, a) - Q(s_{t+N}, a_{t+N}) \right)$$

where $c_0 \coloneqq 1$, and $c_N \coloneqq \prod_{n=0}^{N} \frac{\pi(a_{t+n}|s_{t+n})}{\mu(a_{t+n}|s_{t+n})}$.

Because of c_N, traces are cut when $\pi(a_{t+n}|s_{t+n}) \approx 0$.

UNIVERSITY OF ALBERTA

Conservative Algorithms

- Importance sampling λ-PI
- Retrace (Munos et al., 2016)
- Watkin's Q(λ) (Watkins, 1989)
- Tree-backup (Precup et al., 2000)

Convergence thanks to trace cuts.

Conservative and Non-Conservative Algorithms

UNIVERSITY OF ALBERTA

Conservative Algorithms

- Importance sampling λ-PI
- Retrace (Munos et al., 2016)
- Watkin's Q(λ) (Watkins, 1989)
- Tree-backup (Precup et al., 2000)

Convergence thanks to trace cuts.

Non-conservative Algorithms

- Uncorrected n-step return
- Peng's Q(λ) (Peng & Williams, 1994)
- Harutyunyan's Q(λ) (Harutyunyan et al., 2016)

No generic convergence guarantee.

But some of them are known to yield a better performance.

Research Question

UNIVERSITY OF ALBERTA

Why do non-conservative algorithms outperform theoretically-sound conservative ones?

Theoretical Result

- Convergence guarantee of Peng's Q(λ) with fixed μ
  - Fast convergence (to a biased optimal solution)
  - More robustness to error (compared to VI, λ-PI)
- Convergence guarantee of Peng's Q(λ) with changing μ
  - Convergence under appropriate conditions on μ
  - Slower convergence (to the optimal solution)
  - Less robustness to error (compared to VI, λ-PI)

Empirical Result