UCI University of California, Irvine

CS 277: Control and Reinforcement Learning

Winter 2024

Lecture 17: Offline RL

Roy Fox
Department of Computer Science
School of Information and Computer Sciences
University of California, Irvine

Logistics

assignments

- Exercise 4 due next Monday
- Quiz 8 due next Wednesday
- Exercise 5 will be due Week 11

evaluations

- Course evaluations due next weekend

Today's lecture

The offline setting

Offline policy evaluation

Offline RL

The Bitter Lesson

small data ⟷ big data

- Particularly prevalent in RL
  - Lifelong / continual learning
- We tried hard to be data efficient
  - MBRL, Bounded RL, structure
- "The Bitter Lesson" [Sutton, 2019]
  - In the end, data+compute win

- Web-scale data has huge impact
  - Vision, language, speech, ...
- Why not control?
  - Sparse rewards? Exploration? sure
  - But mostly: big diverse state space
  - More than in language? likely, yes

Why we need on-policy data

on-policy off-policy offline

- Policy-based methods tend to be on-policy

$$\nabla_\theta J_\theta = \mathbb{E}_{s,a\sim p_\pi}[R_{s,a} \nabla_\theta \log \pi_\theta(a \,|\, s)]$$

  - Estimating the gradient by sampling a different distribution is biased

- Value-based methods tend to be off-policy

$$L_\theta(s,a) = (r + \gamma \max_{a'} Q_{\bar{\theta}}(s',a') - Q_\theta(s,a))^2$$

  - Optimally, $L \equiv 0$; but if L is low on train distribution, it may still be high in test

How off-policy can we go?

on-policy ⟵⟶ off-policy ⟵⟶ offline

- Roughly, on-policy loss on off-policy data is incorrect per point
  - We need to collect experience from current policy ⇒ usually small data
  - We can go a tiny bit off-policy, e.g. when parallelizing policy updates
- Off-policy loss on off-policy data is incorrect in expectation
  - We can go significantly off-policy for a long while
  - But in the end, we still need to mitigate the train–test distribution mismatch
  - E.g. by converging toward on-policy experience

What goes wrong without deployment?

- In Offline RL, we get a big experience data but perhaps can't collect more
  - Must operate under severe train–test mismatch $\pi_D \iff \pi_\theta$
  - Better: RL with limited number of deployments
  - Worse: we may not even know π_D; even worse: we may not even see the actions
- The problem is not just that Q may be wrong out-of-distribution (OOD)
  - Policy optimization seeks those states where Q happens to be overestimated
  - Without deployment, we never find out!

$\mathbb{E}_Q[\max_a Q(s,a)] \geq \max_a \mathbb{E}_Q[Q(s,a)]$

winner's curse

What can we do?

- Imitation Learning
  - Missing out on reward signal
- Policy evaluation
  - More advanced, aggressive Importance Sampling techniques
- Policy optimization
  - Constrain π_θ to be close to π_D
  - Constrain the action space to the support of the data
  - Penalize Q_θ outside the support of the data

Today's lecture

The offline setting

Offline policy evaluation

Offline RL

Offline policy evaluation

- We can evaluate off-policy: $Q(s,a) \to r + \gamma \max_{a'} Q(s',a')$
  - This is "robust" to not knowing π_D, but sensitive to errors in Q (can be bad OOD)
- We could also estimate with IS: $\mathbb{E}_{\xi \sim p_\pi}[R(\xi)] = \mathbb{E}_{\xi \sim D}[\rho_D^\pi(\xi) R(\xi)]$
  - This is "robust" to Q errors but sensitive to weight errors $\rho_D^\pi(\xi) = \prod_t \frac{\pi(a_t | s_t)}{\pi_D(a_t | s_t)}$
- Can we be robust to either?

Doubly robust offline RL

- Estimate $V^\pi(s) = \mathbb{E}_{a|s\sim\pi}[Q^\pi(s,a)]$, guess π_D, and sample $(s,a,r,s') \sim D$

$$\hat{V}(s) \to V^\pi(s) + \rho_D^\pi(a\,|\,s)(r + \gamma\hat{V}(s') - Q^\pi(s,a))$$

- If V^π and Q^π are correct then $Q^\pi(s,a) = r(s,a) + \gamma\mathbb{E}_{s'|s,a\sim p}V^\pi(s')$
  - The only consistent solution is $\hat{V} = V^\pi$
- If π_D is correct: $\mathbb{E}_{a,s'\sim\pi_D,p}[\text{RHS}] = \cancel{V^\pi(s)} + \mathbb{E}_{a\sim\pi}[r(s,a) + \gamma\mathbb{E}_{s'\sim p}[\hat{V}(s')] - \cancel{Q^\pi(s,a)}]$
  - Which is TD policy evaluation of π
- Estimator is consistent in either case, but very high variance (have ways to improve)

GenDICE

- $\rho_D^\pi(\xi)$ has high variance, can we do better?

- A different IS: $J_\theta = \mathbb{E}_{s,a\sim p_\pi}[r(s,a)] = \mathbb{E}_{s,a\sim D}[\rho_D^\pi(s,a)r(s,a)]$

  $t \sim \text{Geom}(1-\gamma)$

  - How to find $\rho_D^\pi(s,a) = \dfrac{p_\pi(s,a)}{p_D(s,a)}$?

- Idea: solve consistency recursion

$$\underbrace{p_D(s',a')\rho(s',a')}_{p_\pi(s',a')} = (1-\gamma)\underbrace{p_0(s')}_{p(t=0)}\pi(a'|s') + \underbrace{\gamma\sum_{s,a}}_{p(t)\to p(t+1)}\underbrace{p_D(s,a)\rho(s,a)}_{p_\pi(s,a)}p(s'|s,a)\pi(a'|s')$$

  - Complicated to solve, can be degenerate, but has decent statistical properties

Today's lecture

The offline setting

Offline policy evaluation

Offline RL

Policy constraining

- π shouldn't be far from π_D, there's no data there $\Rightarrow$ constrain $\mathbb{D}[\pi || \pi_D]$
  - Bounded RL: $\max_\pi \mathbb{E}_{s,a\sim p_\pi}[r(s,a)] - \tau \mathbb{D}[\pi || \pi_D]$
- We can use any Bounded RL algorithm, e.g. SAC
  - SAC is off-policy = unbiased per-batch objective, biased expectation
    - $\Rightarrow$ the critic can overestimate the value of π
  - Requires π_D

Implicit policy constraining (e.g. AWR)

- Imagine that we know the bounded-optimal policy

normalizer: $\mathbb{E}_{a|s\sim\pi_D}[\exp(\beta Q_\beta(s,a))]$

$$\pi^* = \arg\max_\pi \mathbb{E}_{s,a\sim p_\pi}[r(s,a)] - \tau\mathbb{D}[\pi\|\pi_D] \propto \pi_D(a|s)\exp(\beta Q_\beta(s,a))$$

imagine demonstrations by π_D

- Then optimize $\arg\min_\pi \mathbb{E}_{s\sim D}[\mathbb{D}[\pi_s^*\|\pi_s]] = \arg\max_\pi \mathbb{E}_{s,a\sim D,\pi^*}[\log\pi(a|s)]$

with actions corrected by π^*

  - This is "Imitation Learning" of the implicit π^*, a.k.a. distillation (BC of known policy)

no need to know π_D

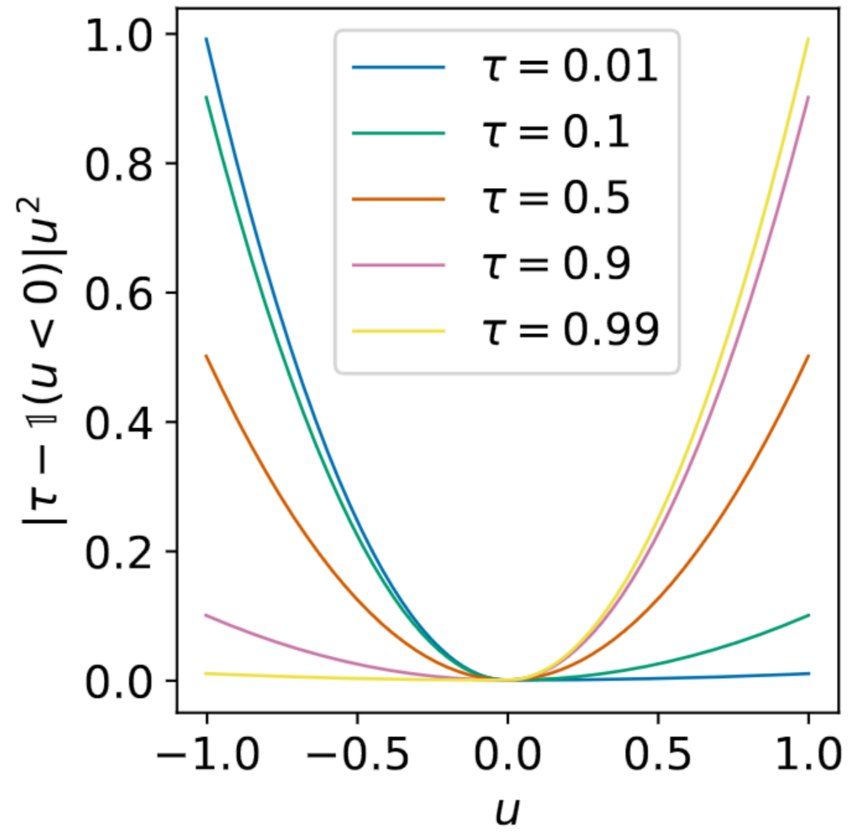
$$\mathbb{E}_{s,a\sim D,\pi^*}[\log\pi(a|s)] = \mathbb{E}_{s,a\sim D}[\exp(\beta A_\beta(s,a))\log\pi(a|s)]$$

  - with $A_\beta(s,a) = Q_\beta(s,a) - V_\beta(s) = Q_\beta(s,a) - \frac{1}{\beta}\log\mathbb{E}_{a|s\sim\pi_D}[\exp(\beta Q_\beta(s,a))]$

- Essentially, supervised learning with NLL, weighted by $\exp(\beta A_\beta(s,a))$

Implicit Q-Learning (IQL)

- Bounded RL constrains π to be close to π_D
  - If $\pi_D(a \mid s)$ is small but we sampled it enough, still hard to diverge to large $\pi(a \mid s)$
- Instead, allow π to diverge freely over well-supported actions



- Expectile: $\ell^\tau(u) = \frac{1}{2}u^2 + (\tau - \frac{1}{2}) \, |u| \cdot u$

$$V \to \arg\min_V \mathbb{E}_{s,a \sim D}[\ell^\tau(Q(s,a) - V(s))] \qquad Q \to r + \gamma V$$

  - As $\tau \to 1$, $V(s)$ will match $\max_a Q(s,a)$ for rarer and rarer greedy actions in D

Conservative Q-Learning (CQL)

- Perhaps we can tackle the problem more directly
  - If the issue is that Q can be overestimated OOD, let's penalize it OOD

$$L_\theta(s, a, r, s' \sim D) = (r + \gamma \mathbb{E}_{a'|s' \sim \pi}[Q_{\bar{\theta}}(s', a')] - Q_\theta(s, a))^2 + \lambda \mathbb{E}_{\tilde{a}|s \sim \pi}[Q_\theta(s, \tilde{a})]$$

  - For large enough λ, L_θ is minimized for conservative $Q_\theta \leq Q^\pi$
- But this also underestimates Q in-distribution
  - Subtract a loss term $\lambda Q_\theta(s, a)$ to not penalize in-distribution
  - Now $V_\theta = \mathbb{E}[Q_\theta]$ is conservative, but Q_θ may not be

Recap

- It'd be nice to use web-scale data, but it's offline
  - Maybe even with an unknown policy / actions
  - But there may be no better way to get RL foundation models
- Optimize under uncertainty ⇒ tend to overestimate (winner's curse)
- In Online RL (On/Off-Policy), we overcome this by collecting more data
- In Offline RL, we overcome this through
  - Aggressive Importance Sampling ⇒ can be high variance
  - Constraining our solution to the support of the data ⇒ can't improve much

without further assumptions / prior knowledge