DS 598
Introduction to RL

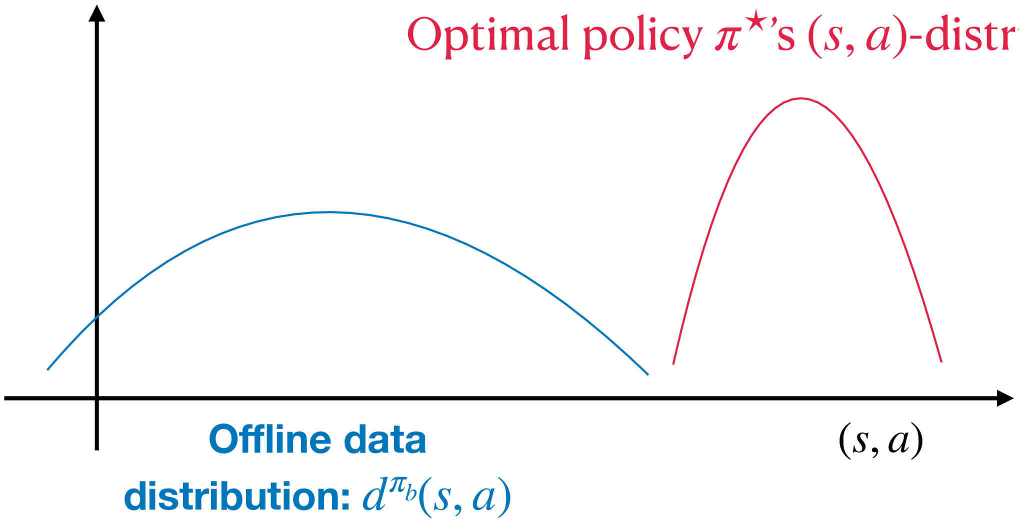
Xuezhou Zhang

Chapter 9: Offline RL (Continued)

Offline Data Coverage

$$d^{\pi_b} \in \Delta(S \times A)$$

$$\mathscr{D} = \{s, a, s'\}, \text{ where } s, a \sim d^{\pi_b}, s' \sim P(\cdot \mid s, a)$$



Finding $\pi^\star$ seems hopeless!

Constrained Pessimistic Policy Optimization (CPPO)

1. MLE: $\hat{P} = \max_{P \in \mathcal{P}} \sum_{s,a,s' \in \mathcal{D}} \ln P(s' \mid s, a)$

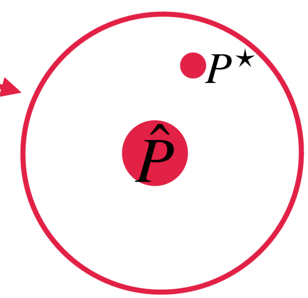
2. Constrained Pessimistic Policy Optimization

$$\max_{\pi} \min_{P \in \mathcal{P}} J(\pi; P)$$

Select the least favorable model!

$$\text{s.t.,} \frac{1}{|\mathcal{D}|} \sum_{s,a \in \mathcal{D}} \left\| P(\cdot \mid s, a) - \hat{P}(\cdot \mid s, a) \right\|_1 \leq \delta$$

$$\left(\text{or } \frac{1}{|\mathcal{D}|} \sum_{s,a,s' \in \mathcal{D}} \ln P(s' \mid s, a) \geq \frac{1}{|\mathcal{D}|} \sum_{s,a,s' \in \mathcal{D}} \ln \hat{P}(s' \mid s, a) - \delta \right)$$

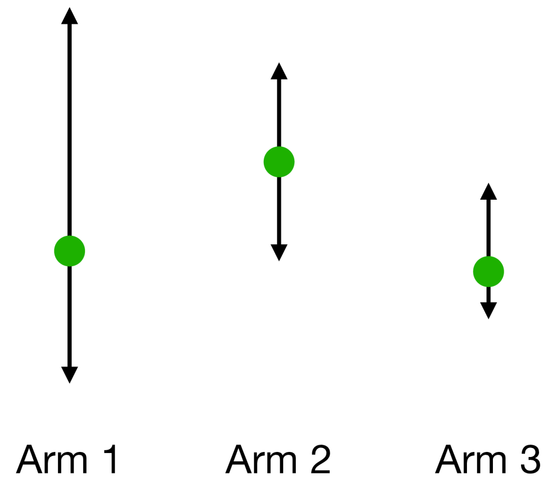


Pessimism seems key in achieving robustness.

🤔 Can we get it without solving a constrained optimization problem?

Recap:

Multi-armed Bandits and UCB Algorithm



$$a^n := \arg\max_a \{\hat{\mu}^n(a) + \sqrt{\ln(KN/\delta)/N^n(a)}\}$$

$$\mathbb{E}\left[N\mu(a^\star) - \sum_{n=1}^{N} \mu(a^n)\right] \leq \widetilde{O}(\sqrt{KN})$$

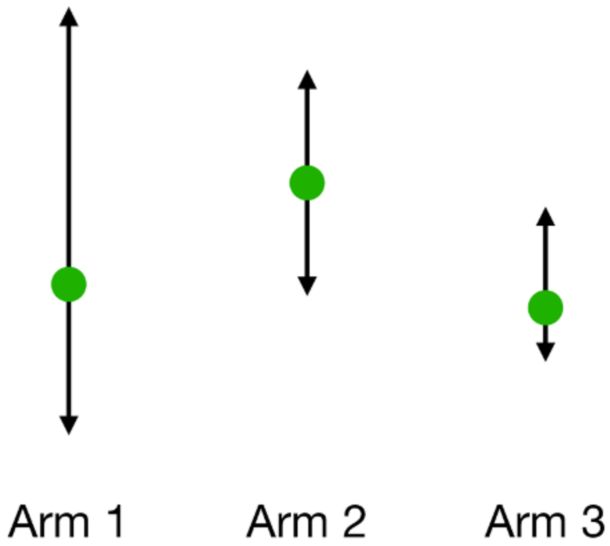
Key step in the proof:

$$\mu(a^\star) - \mu(a^n) \leq \widehat{\mu}(a^n) + \sqrt{\frac{\ln(KN/\delta)}{N^n(a_n)}} - \mu(a^n)$$

"optimism in the face of uncertainty (OFU)"

What if, instead of adding the UCB bonus, we subtract it?

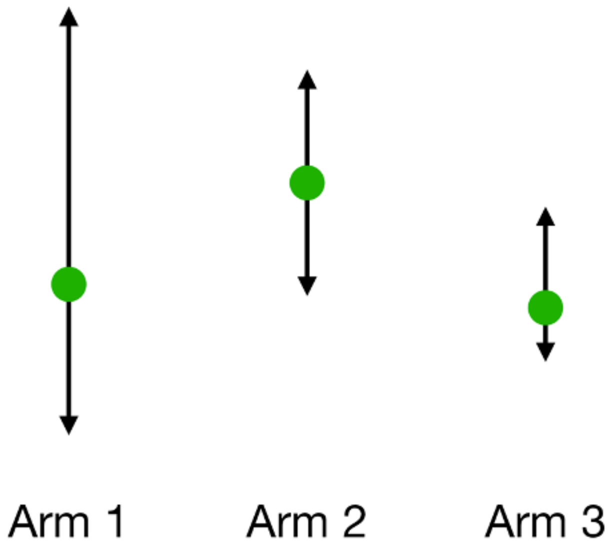
The Lower-Confidence Bound Algorithm?



$$\hat{a} := \mathrm{argmax}_a \hat{\mu}(a) - \sqrt{\ln(KN/\delta)/N(a)}$$

What can we achieve?

The Lower-Confidence Bound Algorithm?



$$\hat{a} := \text{argmax}_a \hat{\mu}(a) - \sqrt{\ln(KN/\delta)/N(a)}$$

What can we achieve?

Against any comparator arm a, the arm $\hat{a}$ we pick will have a reward at least

$$\mu(a) - \mu(\hat{a}) \leq \sqrt{\ln\left(\frac{KN}{\delta}\right)/N(a)}$$

"pessimism in the face of uncertainty (OFU)"

Formal Theoretical Guarantee for CPPO

2. CPPO's Sample Complexity:

Given n (i.i.d) offline data points, with high probability:

$$\forall \pi^*; V_{P^\star}^{\pi^*} - V_{P^\star}^{\hat{\pi}} = O\left(H^2 \sqrt{\frac{C_{\pi^*}^{\dagger} \ln(|\mathscr{P}|/\delta)}{n}} \right)$$

In the bandit setting: $C_{\pi^*}^{\dagger} = \sup_{s,a} \frac{d^{\pi}(s,a)}{d^{\pi_b}(s,a)} = 1/d^{\pi_b}(a)$

LCB achieves the same effect as Constrained Policy Optimization!

UCBVI: Optimistic Model-based Learning

Inside iteration n :

Use all previous data to estimate transitions $\widehat{P}^n$

Design reward bonus $b_h^n(s, a), \forall s, a, h$

Optimistic planning with learned model: $\pi^n = \text{Value-Iter}\left(\widehat{P}^n, \{r_h + b_h^n\}_{h=1}^{H-1} \right)$

Collect a new trajectory by executing π^n in the real world P starting from s_0

LCBVI: Pessimistic Model-based Learning

~~UCBVI: Optimistic Model-based Learning~~

Inside iteration n :

Use all previous data to estimate transitions $\widehat{P}^n$

Design reward bonus $b_h^n(s,a), \forall s,a,h$

$\{r_h - b_h\}_{h=1}^{H-1}$

Optimistic planning with learned model: $\pi^n = \text{Value-Iter}\left(\widehat{P}^n, ~~\{r_h + b_h^n\}_{h=1}^{H-1}~~\right)$

Collect a new trajectory by executing π^n in the real world P starting from s_0

LCBVI achieves the same type of guarantee as CPPO!

One of the most important observations in RL:

The symmetry between
online (optimism) and offline (pessimism)
learning

- Any reward bonus-type exploration mechanism can be immediately turned to a robust-learning mechanism in offline RL.

- Psuedo-based bonus
- Hashmap-based bonus
- Uncertainty-estimation
- Random Network Distillation (RND)
- …

- All you need to do in your code: change the “+” sign to “-”

Some other approaches from the Empirical Community:

1. **KL regularization:** $\hat{\boldsymbol{\pi}} = \mathbf{argmax}\, \boldsymbol{J_D}(\boldsymbol{\pi}) + \boldsymbol{\alpha} * \mathbf{KL}(\boldsymbol{\pi}|\boldsymbol{\pi_b})$

(Requires the knowledge of the data collecting policy)

Equivalent to running TRPO/PPO on the offline data and use π_b as the reference policy to calculate the regularizer.

This is how ChatGPT is trained!

Some other approaches from the Empirical Community:

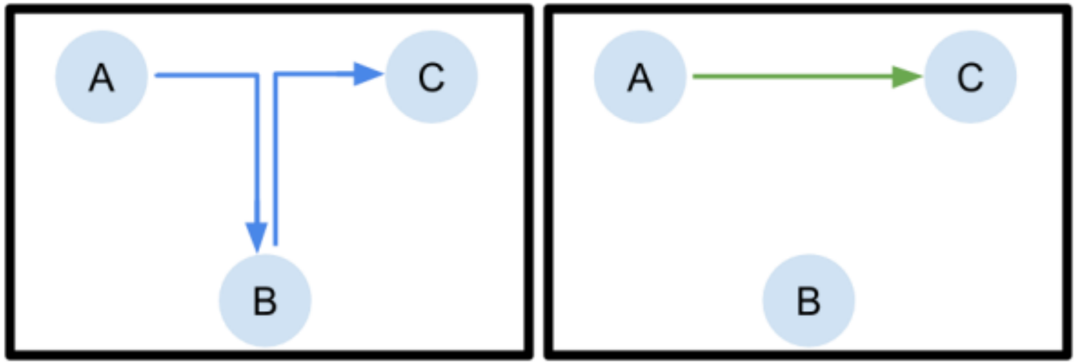
1. **KL regularization:** $\hat{\pi} = \text{argmax}\, J_D(\pi) + \alpha * \text{KL}(\pi|\pi_b)$

✓ Pro: able to regularize the learned policy.

✓ Pro: Extremely easy to implement

Con: Can't realize the full potential of the offline data.

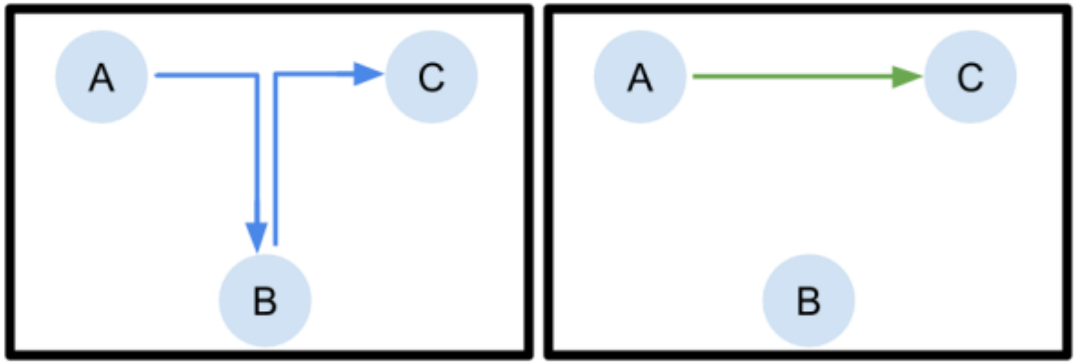
Recall the "stitching" effect:



Some other approaches from the Empirical Community:

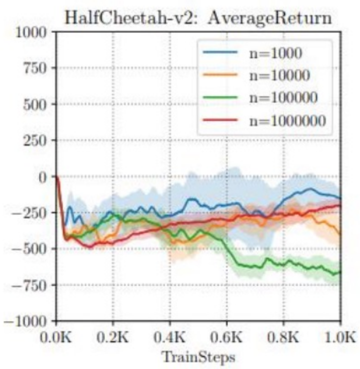
1. **KL regularization:** $\hat{\pi} = \text{argmax}\, J_D(\pi) + \alpha * \text{KL}(\pi|\pi_b)$

- This is also called advantageous imitation learning:
  - The KL term alone would be imitation learning
- The first term tries to improve upon the behavior policy in a KL-restricted neighborhood.

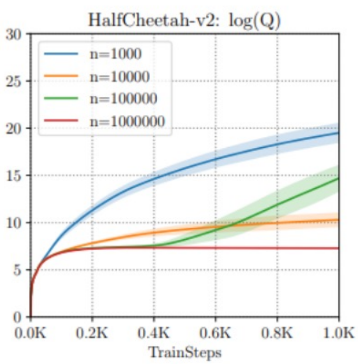


Some other approaches from the Empirical Community:

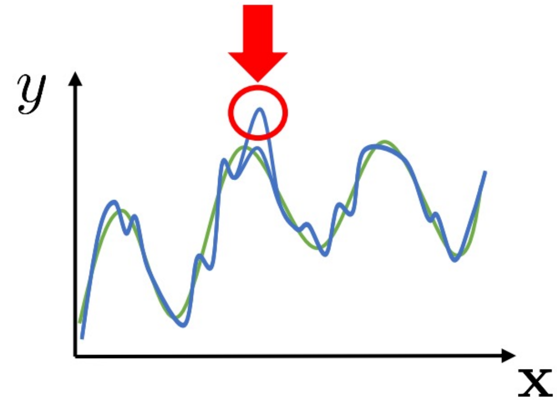
2. Conservative Q-learning (CQL)



how well it does

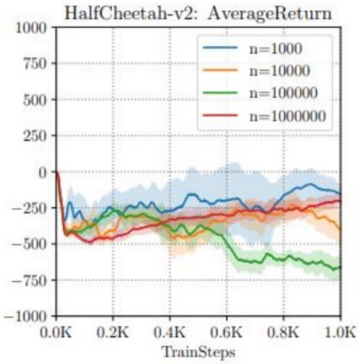


how well it *thinks* it does (Q-values)

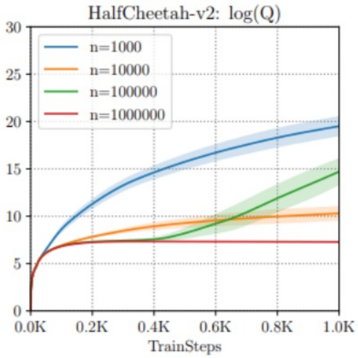


Some other approaches from the Empirical Community:

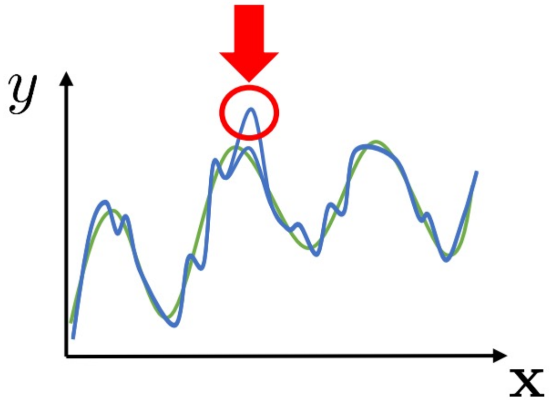
2. Conservative Q-learning (CQL)



how well it does



how well it *thinks* it does (Q-values)

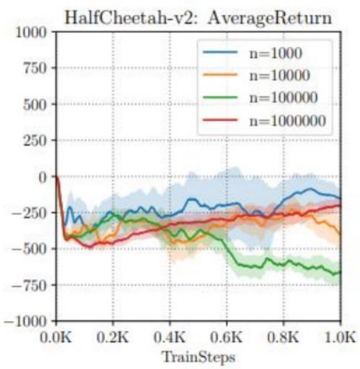


$$\hat{Q}^{\pi} = \arg\min_{Q} \max_{\mu} \alpha E_{\mathbf{s}\sim D, \mathbf{a}\sim\mu(\mathbf{a}|\mathbf{s})}[Q(\mathbf{s},\mathbf{a})]$$ term to push down big Q-values

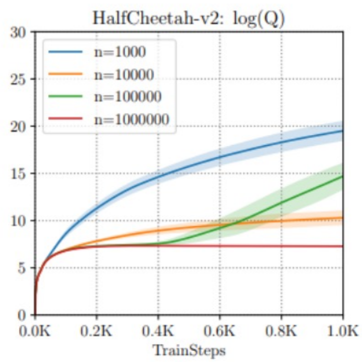
regular objective $$+E_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim D}\left[(Q(\mathbf{s},\mathbf{a}) - (r(\mathbf{s},\mathbf{a}) + E_{\pi}[Q(\mathbf{s}',\mathbf{a}')]))^2\right]$$

Some other approaches from the Empirical Community:

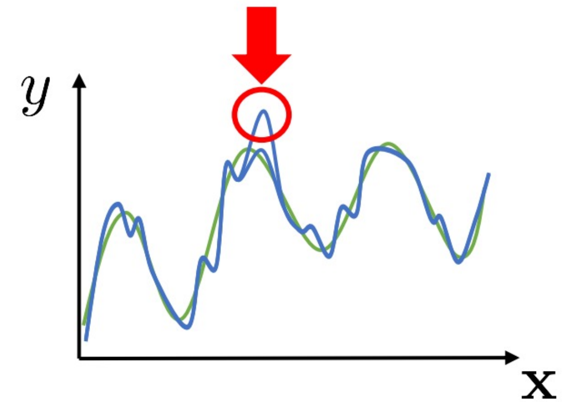
2. Conservative Q-learning (CQL)



how well it does



how well it *thinks* it does (Q-values)



$$\hat{Q}^\pi = \arg\min_Q \max_\mu \alpha E_{\mathbf{s}\sim D,\mathbf{a}\sim\mu(\mathbf{a}|\mathbf{s})}[Q(\mathbf{s},\mathbf{a})]$$ } term to push down big Q-values

regular objective { $$+E_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim D}\left[(Q(\mathbf{s},\mathbf{a}) - (r(\mathbf{s},\mathbf{a}) + E_\pi[Q(\mathbf{s}',\mathbf{a}')]))^2\right]$$

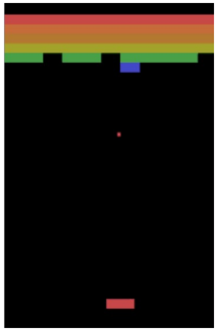
can show that $\hat{Q}^\pi \leq Q^\pi$ for large enough α

↑ true Q-function

Some other approaches from the Empirical Community:

3. **There are many more…**

Is that all?

$($  $,$  $)_{1:M}$

- Given a dataset of transition $D = \{(s_t, a_t, s'_t, r_t)\}_{t=1:T}$.
- Find the "best possible" policy π_θ.

🤔 Is this really the right objective?