CS292F StatRL Lecture 11
Exploration in Linear MDP & Introduction to offline RL

Instructor: Yu-Xiang Wang

Spring 2021

UC Santa Barbara

Logistics

- Project midterm milestone due
  - Important as I need to allocate space for student presentation
- For those who haven’t submitted HW1
  - You don’t have to solve everything, just submit what you have
  - HW1 is long I am thinking of adjusting grading criteria
- HW2 is not as long
  - Don’t wait

Recap: Lecture 10

- Exploration in Linear MDPs
- Properties of Linear MDPs
- Algorithm: UCB-VI for Linear MDPs
- Regret analysis

Recap: Impossibility results

- What are the assumptions to make?
  - **$Q^*(s,a)$ approximately linear?**
  - **$Q^\pi(s,a)$ is approximately linear for all π?**
  - $Q^*(s,a)$ is exactly linear?
  - $Q^\pi(s,a)$ is exactly linear for all π?

Weisz et al (ALT-2020): http://proceedings.mlr.press/v132/weisz21a.html

Exponential sample complexity / regret lower bounds for the approximate case…

(Du, Kakade, Wang, Yang, 2019) Is a good representation sufficient for sample efficient reinforcement learning?

Recap: Linear MDPs

- Exists feature map $\phi : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}^d$
  - Such that:

$$r_h(s,a) = \theta_h^\star \cdot \phi(s,a), \quad P_h(\cdot|s,a) = \mu_h^\star \phi(s,a), \forall h$$

(Jin et al., 2020) Provably efficient reinforcement learning with linear function approximation

Recap: UCB-VI for Linear MDPs

- In every round:
  1. Run Ridge regression for estimating the model

$$\widehat{\mu}_h^n = \mathrm{argmin}_{\mu \in \mathbb{R}^{|S| \times d}} \sum_{i=0}^{n-1} \left\| \mu \phi(s_h^i, a_h^i) - \delta(s_{h+1}^i) \right\|_2^2 + \lambda \|\mu\|_F^2.$$

$$\widehat{\mu}_h^n = \sum_{i=0}^{n-1} \delta(s_{h+1}^i) \phi(s_h^i, a_h^i)^\top (\Lambda_h^n)^{-1}$$

  2. Construct the exploration bonuses

$$b_h^n(s,a) = \beta \sqrt{\phi(s,a)^\top (\Lambda_h^n)^{-1} \phi(s,a)},$$

  3. Run optimistic value iterations, and update greedy policy

Recap: Regret bound

- Choose $$\beta = Hd\left(\sqrt{\ln\frac{H}{\delta}} + \sqrt{\ln(W+H)} + \sqrt{\ln B} + \sqrt{\ln d} + \sqrt{\ln N}\right)$$

$$\lambda = 1$$

$$b_h^n(s,a) = \beta\sqrt{\phi(s,a)^\top(\Lambda_h^n)^{-1}\phi(s,a)},$$

- Regret $$\tilde{O}\left(H^2\sqrt{d^3N}\right)$$

Recap: Regret analysis

- Regret of episode t

Per-episode Regret:

$$V^* - V^{\pi_n} = V_0^*(s_0) - V_0^{\pi_n}(s_0) \overset{\text{optimism}}{\le} \hat{V}_0^{\pi_n}(s_0) - V_0^{\pi_n}(s_0)$$

Simulation Lemma $\rightarrow$

$$\le \sum_{h=0}^{H-1} E^{\pi_n}\left[b_h^n(s_h,a_h) + \left(\hat{P}_h^n(\cdot|s_h,a_h) - P_h(\cdot|s_h,a_h)\right)\cdot \hat{V}_{h+1}^n \right]$$

$$\le \sum_{h=0}^{H-1} E^{\pi_n}\left[b_h^n(s_h,a_h) \right]$$

  - Optimism / simulation lemma

- Sum them up to get total regret

Lemma (Information Gain bound)

$\forall\, s_h^n, a_h^n$ sequence $\quad \sum_{n=0}^{N-1} \phi(s_h^n,a_h^n)^\top (\Lambda_h^n)^{-1} \phi(s_h^n,a_h^n) = \tilde{O}(d \log N)$

  - Same information-gain bound from linear bandits

Recap: It remains to prove

- 1. Uniform convergence bound

- 2. “Optimism”

The same induction argument as in the UCB-VI for tabular MDP
(Read Lemma 7.10 in AJKS)

- 3. “Information gain” bound

The same argument as in the Linear Bandits case.
(Read Lemma 7.12 in AJKS)

Recap: Bound for a fixed V

- Lemma 7.3 AJKS

$$\widehat{\mu}_h^n - \mu_h^\star = -\lambda \mu_h^\star \left(\Lambda_h^n\right)^{-1} + \sum_{i=1}^{n-1} \epsilon_h^i \phi(s_h^i, a_h^i)^\top \left(\Lambda_h^n\right)^{-1}.$$

- The quantity of interest is a inner product with this:

$$\left[\left(\hat{\mu}_h^n - \mu_h^n \right) \cdot \phi(s,a) \right)^\top V = \phi(s,a)^\top \underbrace{\left(\hat{\mu}_h^n - \mu_h^n \right)^\top \cdot V}$$

Challenge: we cannot use union bound because we have an infinite number of value functions

- A covering number argument.

- Covering number: the number of balls with radius ε that is needed to cover all points in a set.

Family of value functions we consider

$$f_{w,\beta,\Lambda}(s) = \min\left\{\max_a \left(w^\top \phi(s,a) + \beta\sqrt{\phi(s,a)^\top \Lambda^{-1}\phi(s,a)}\right),\ H\right\}, \forall s \in \mathcal{S}.$$

$$\mathcal{F} = \{f_{w,\beta,\Lambda} : \|w\|_2 \le L, \beta \in [0,B], \sigma_{\min}(\Lambda) \ge \lambda\}.$$

What is a finite set to cover this class such that for every f in this set, there is a function in the finite set, such that they are ε-close in sup-norm?

Covering number calculations

From covering number to a uniform convergence bound

Final notes about linear MDPs

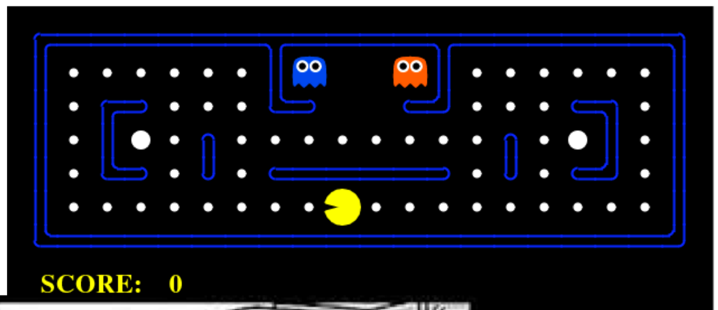
- A semi-parametric model
  - The number of parameters to describe the model can be exponentially large: d S
  - Efficient algorithm with regret independent to S
- Still very strong assumption on the feature map
  - Interesting open problems:
    - Representation learning
    - Nonlinear parametric models
    - Suboptimal rates when naively applying to the tabular case

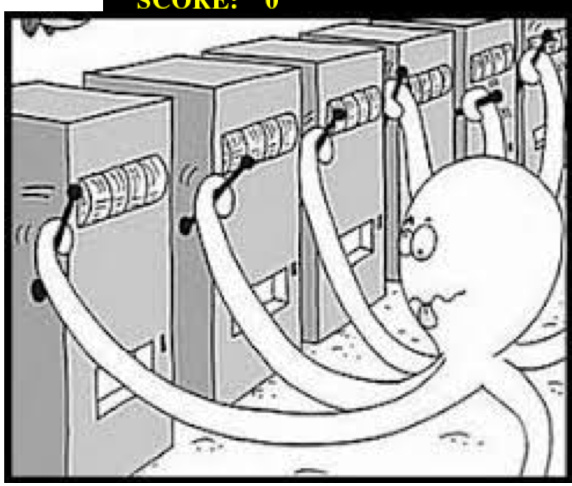
Remainder of the lecture

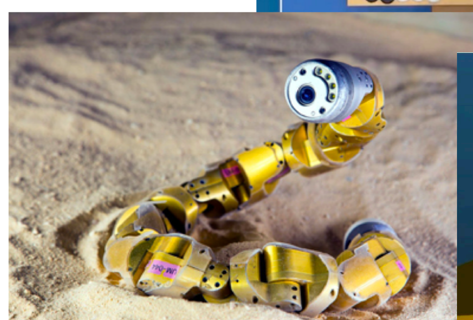
- Introduction to offline reinforcement learning

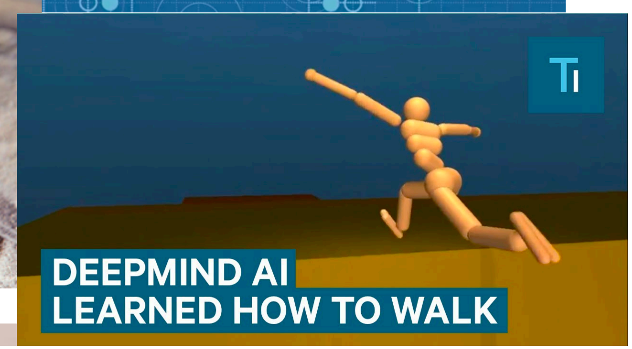
- Off-policy evaluation in contextual bandits

Recap: RL is among the hottest area of research in ML!

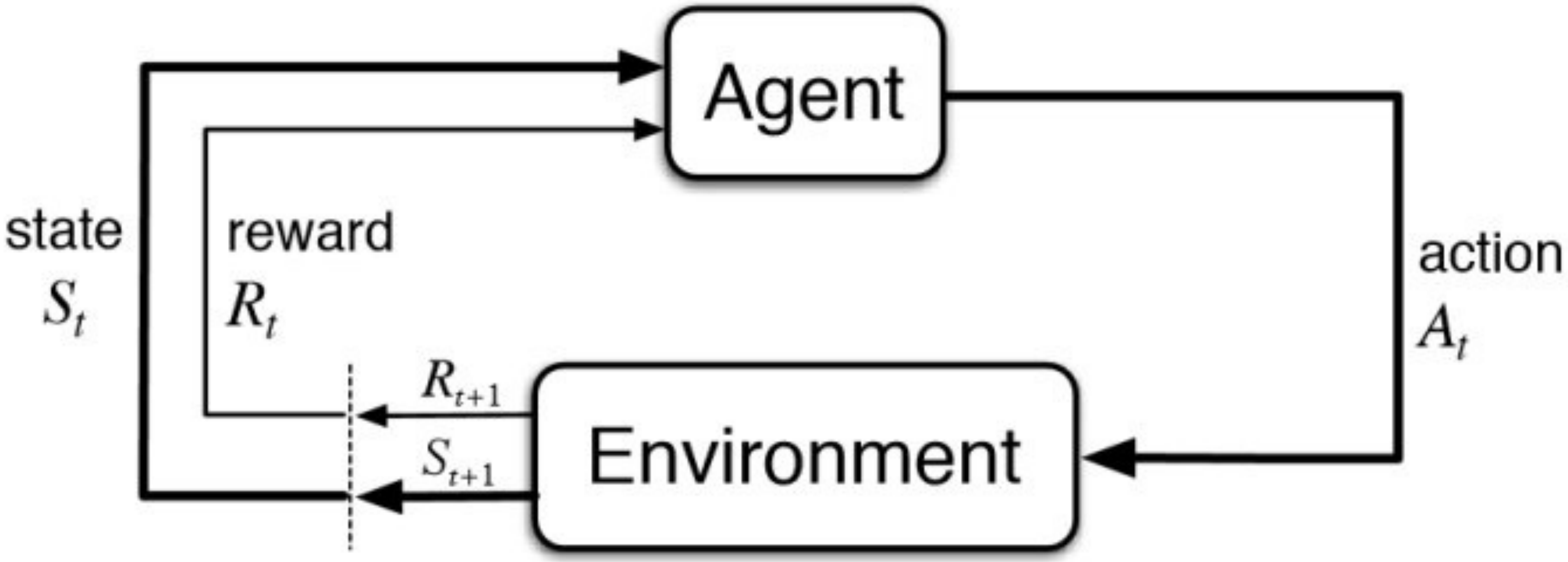








An RL agent learns **interactively** through the **feedbacks** of an environment.



- Learning how the world works (dynamics) and how to maximize the long-term reward (control) at the same time.

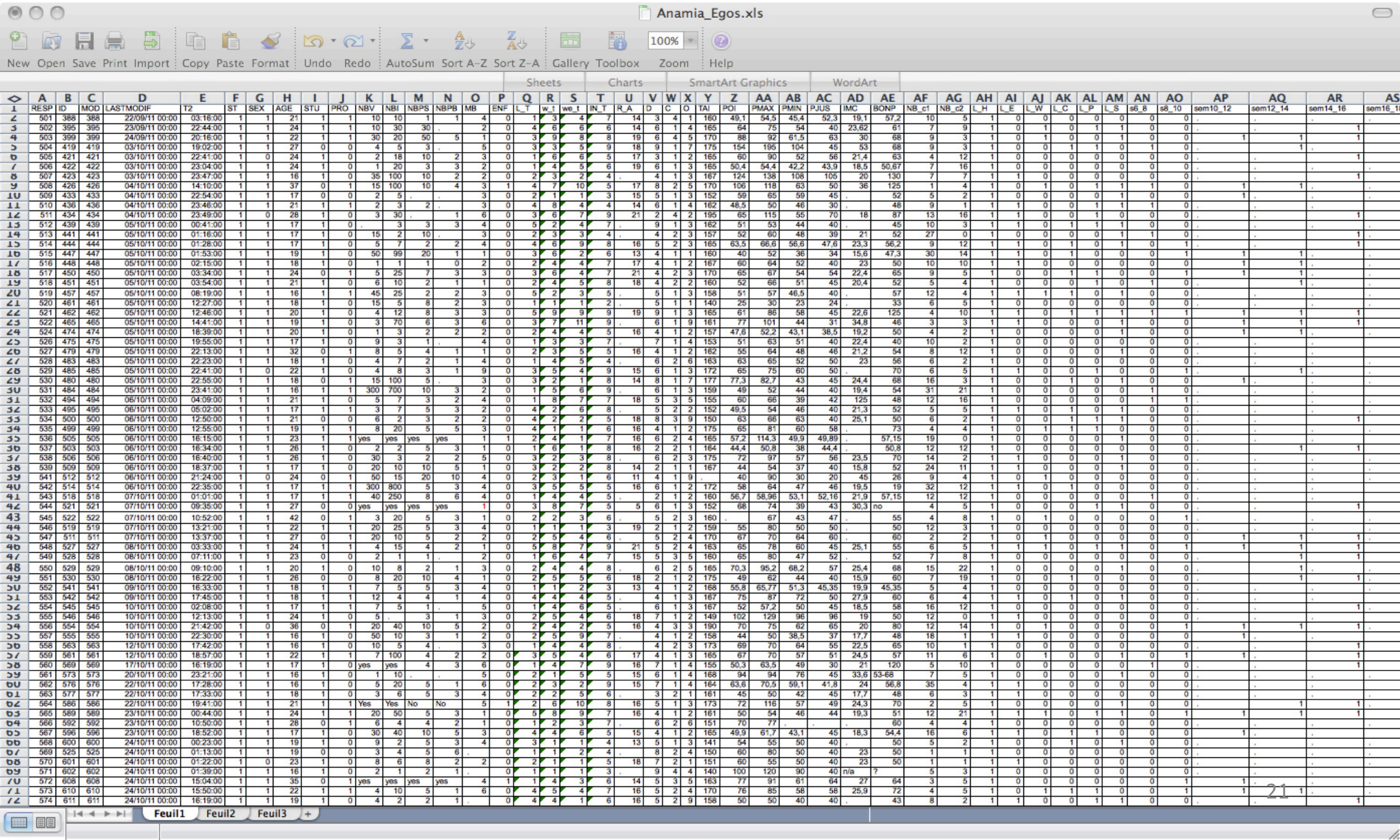
Applications of RL in the real life

- RL for robotics.
- RL for dialogue systems.
- RL for personalized medicine.
- RL for self-driving cars.
- RL for new material discovery.
- RL for sustainable energy.
- RL for feature-based dynamic pricing.
- RL for maximizing user satisfaction.
- RL for QoE optimization in networking
- …

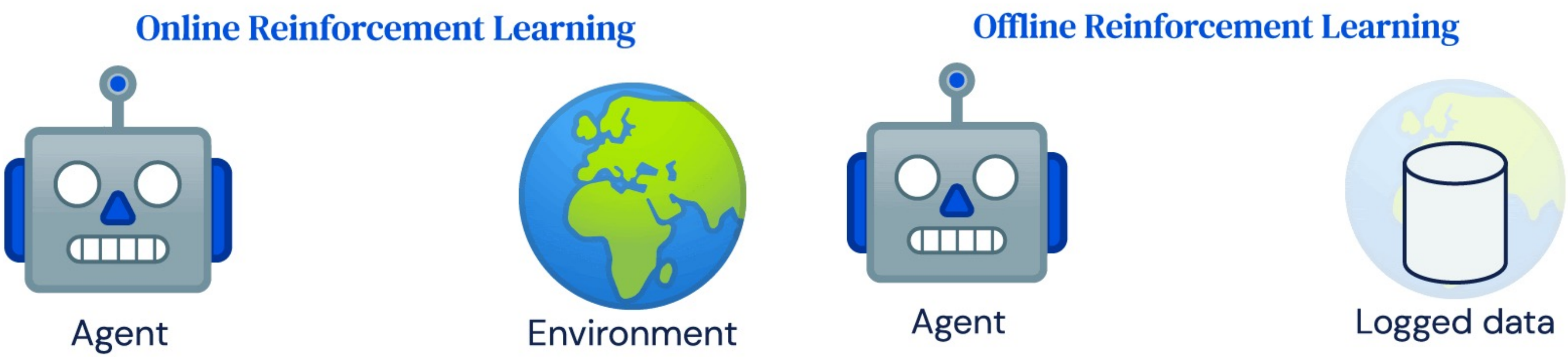
Challenges of Reinforcement in the real life

- No access to a simulator
- Every data point is costly.
- Legal, safety issues associated with exploration.
- Large / complex state-space, action space.
- Long horizon
- Limited adaptivity (cannot run too many iterations)

From an Applied ML Scientist point of view, the starting point of a project is often:



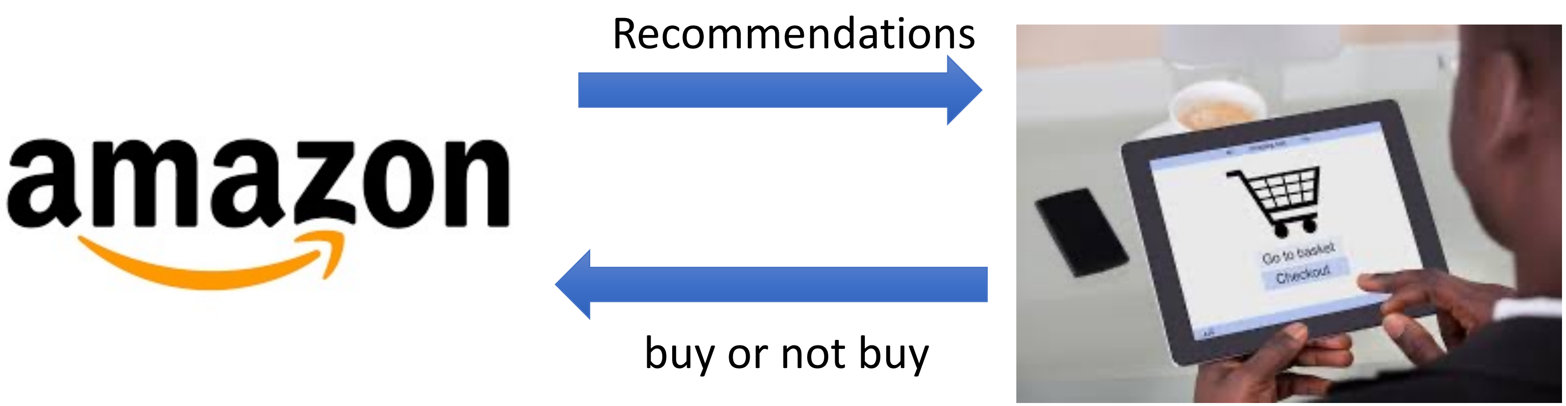
Online RL vs Offline RL



Exploration is often **expensive**, **unsafe**, **unethical** or **illegal** in practice, e.g., in self-driving cars, or in medical applications.

Can we learn a policy from already **logged interaction data**?

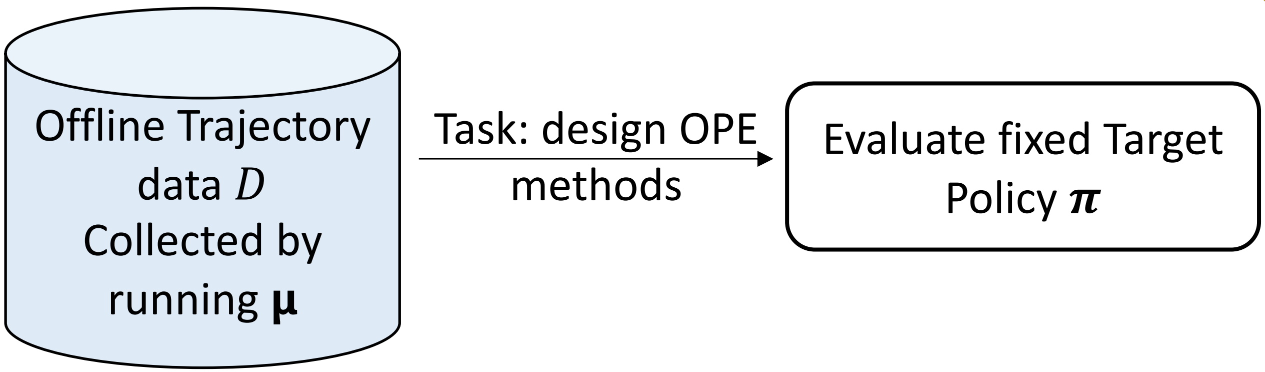
Off-Policy learning: an example



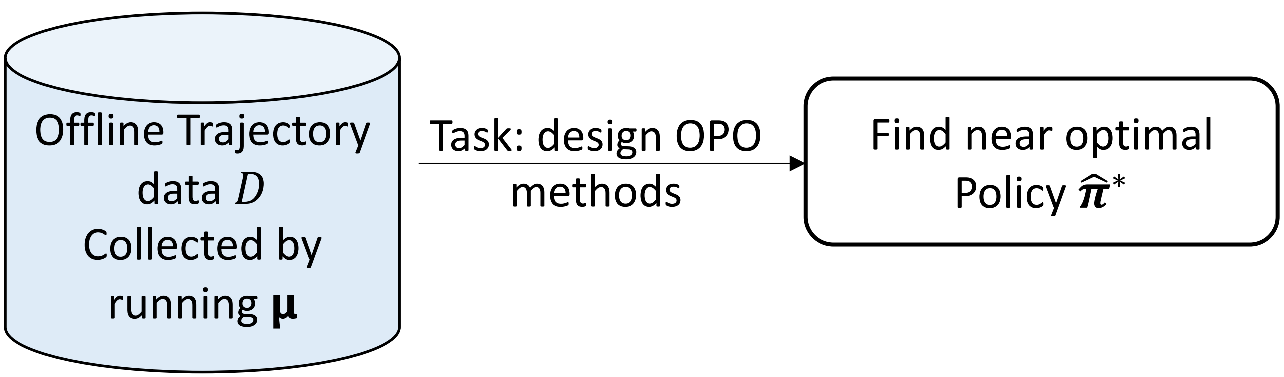
- How to evaluate a new algorithm without actually running it live?
- How to learn a better system than the one that is deployed.

Offline Reinforcement Learning, aka. Batch RL

- Task 1: Offline Policy Evaluation. (OPE)



- Task 2: Offline Policy Learning. (OPL)



Contextual bandits model

- Contexts:
  - $x_1, \ldots, x_n \sim \lambda$ drawn iid, possibly infinite domain
- Actions:
  - $a_i \sim \mu(a|x_i)$ Taken by a randomized "Logging" policy
- Reward:
  - $r_i \sim D(r|x_i, a_i)$ Revealed only for the action taken
- Value:
  - $v^{\mu} = \mathbb{E}_{x\sim\lambda}\mathbb{E}_{a\sim\mu(\cdot|x)}\mathbb{E}_D[r|x,a]$
- We collect data $(x_i, a_i, r_i)_{i=1}^{n}$ by the above processes.

Off-policy Evaluation and Learning

Off-policy evaluation

Estimate the value of a fixed target policy π

$$v_\pi := \mathbb{E}_\pi[\text{Reward}]$$

Off-policy learning

find $\pi \in \Pi$

that maximizes v_π

- Using data $(x_i, a_i, r_i)_{i=1}^n$
- often the policy μ or logged propensities $(\mu_i)_{i=1}^n$

ATE estimation is a special case of off-policy evaluation

- a: Action ⇔ T: Treatment {0,1}
- r: Reward ⇔ Y: Response variable
- x: Contexts ⇔ X: covariates

Direct Method / Regression-estimator

- Fit a regression model of the reward

$$\hat{r}(x, a) \approx \mathbb{E}(r | x, a)$$ using the data

- Then for any target policy

$$\hat{v}^{\pi}_{\mathrm{DM}} = \frac{1}{n} \sum_{i=1}^{n} \sum_{a \in \mathcal{A}} \boxed{\hat{r}(x_i, a)} \pi(a | x_i)$$

Pros:

- Low-variance.
- Can evaluate on unseen contexts

Cons:

- Often high bias
- The model can be wrong/hard to learn

Inverse propensity score / Importance sampling

(Horvitz & Thompson, 1952)

Importance weights

$$\hat{v}^{\pi}_{\text{IPS}} = \frac{1}{n}\sum_{i=1}^{n} \frac{\pi(a_i|x_i)}{\mu(a_i|x_i)} r_i$$

$=: \rho_i$

Pros:

- No assumption on rewards
- Unbiased
- Computationally efficient

Cons:

- High variance when the weight is large

Next lecture: OPE for reinforcement learning

- Importance sampling

- Marginalized importance sampling