Advanced Topics in Reinforcement Learning

Lecture 24: Offline RL

Josiah Hanna
University of Wisconsin — Madison

Announcements

- Thursday: finish offline RL + applications
- Next class: project lightning talks
- Final projects due December 12
- Please complete the course evaluation! At 4% right now

Learning Outcomes

After today, you will be able to:

1. Analyze limitations of applying RL methods to static datasets.
2. Evaluate techniques for addressing offline RL challenges.
3. Identify challenges with evaluating policies from static datasets.

Offline RL Introduction

- Online RL is what we have covered so far.
- Exploration makes online RL slow.
  - Have to collect data before any learning can begin and more data as the policy changes.
  - I.e., (s,a,s’,r) tuples.
- What if we already have data available to us?
  - Offline RL is RL applied to a static dataset of (s,a,s’,r) tuples without additional exploration.
  - Also called “Batch RL.” Batch RL is the older term and offline RL has gained prominence recently along with much attention.

Offline RL Motivation

- Modern machine learning is being driven by 1) enormous data sets and 2) large neural networks.
- Collecting a large data set through task interaction often takes a long time.
- What if we have existing data from:
  - Previously used policies (possibly non-RL policies)?
  - Other tasks?
  - Data from humans (e.g., YouTube videos of people cooking dinner)?
- Many potential applications:
  - Self-driving cars (large amounts of data available).
  - Healthcare (exploration is limited or impossible).
  - Robotics (diverse data available).

Offline RL Formalism

- Assume the target task can be described as an MDP.
  - More on partial observability next class.
- A *behavior policy,* $\pi_\beta(a \mid s)$, has collected dataset $\mathscr{D} = \{(s_i, a_i, s'_i, r_i)\}_{i=1}^{m}$.
  - Possibly multiple behavior policies and possibly unknown to us.
- Goal: Use $\mathscr{D}$ to learn policy, π, that maximizes expected return when deployed on the target task.

Warm-up: Imitation learning

- Given $\mathscr{D}$, we can attempt to just mimic the behavior policy that generated the data.

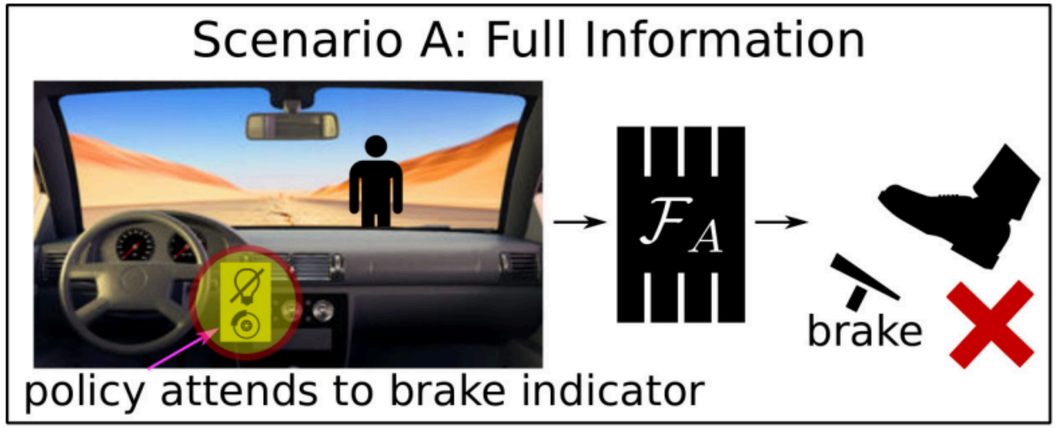
  Supervised learning not reinforcement learning.

  - $$\pi \leftarrow \arg\max_{\pi} \sum_{i=1}^{m} \log \pi(a_i \mid s_i)$$

- (Sort of) robust to distribution shift.

  Performance loss can be quadratic in the episode length.

- Limitations:
  - Cannot improve upon π_β (and may do worse).
  - Causal confusion.



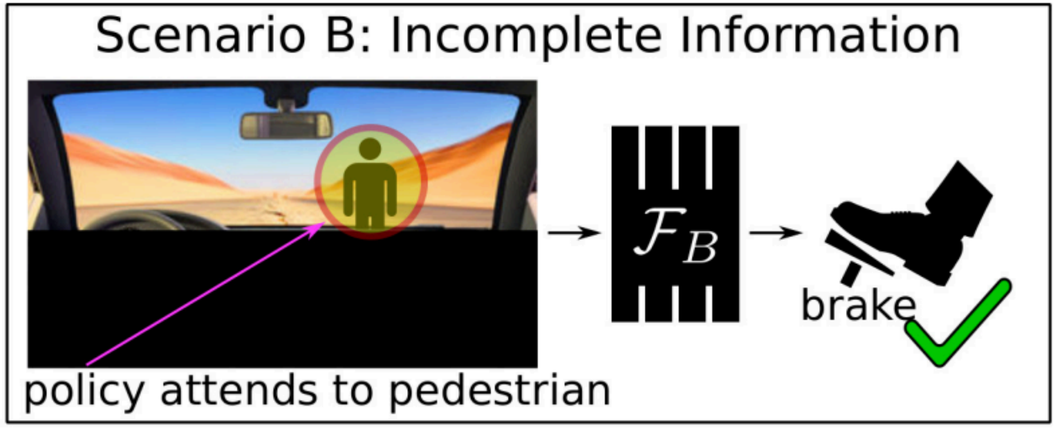


Image credit: Causal Confusion in Imitation Learning. De Haan et al. 2019.

Challenges

- Distribution shift: distribution of data in $\mathscr{D}$ is different than it would be if $\mathscr{D}$ was collected with the current policy, π.
  - Similar challenge for any off-policy RL algorithm but more extreme in offline RL.
- Missing data for some actions.
  - Should we take or avoid those actions?

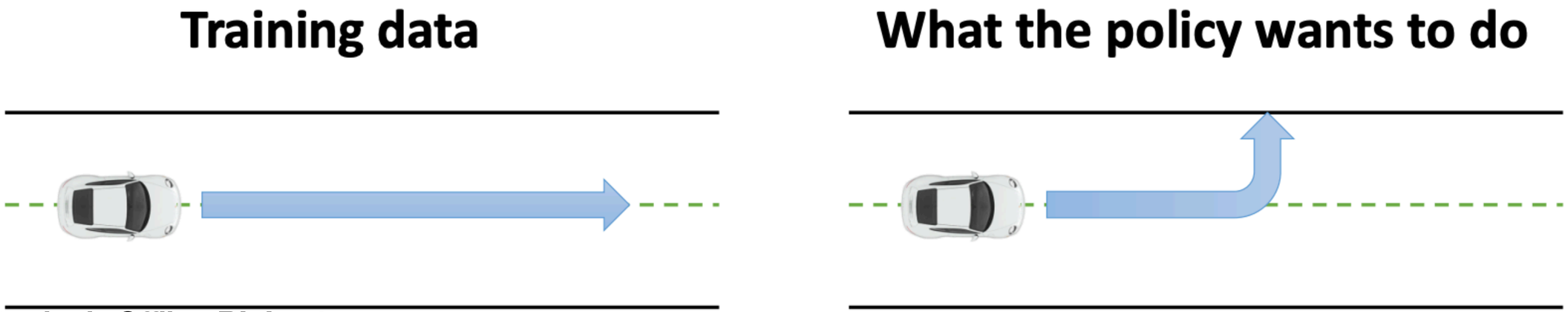


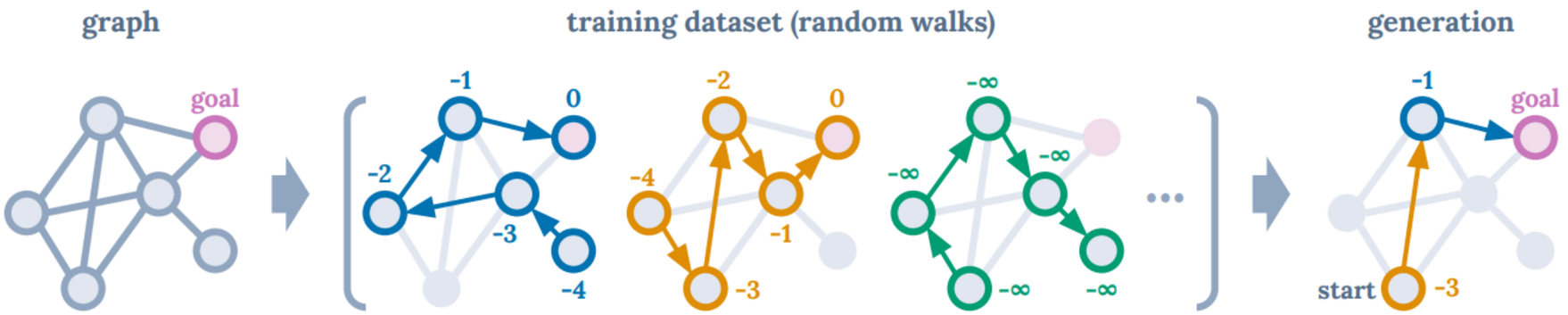
Image credit: Sergey Levine's Offline RL Lecture

Bowman’s Presentation

- Off-policy Deep Reinforcement Learning without Exploration
- Fujimoto et al. 2018
- Slides

What do we want in offline RL?

- Offline RL should improve upon π_β.
- Combine the best parts of sub-optimal behaviors.



Decision Transformer: Reinforcement Learning via Sequence Modeling. Chen et al. 2021.

Offline RL Method Classes

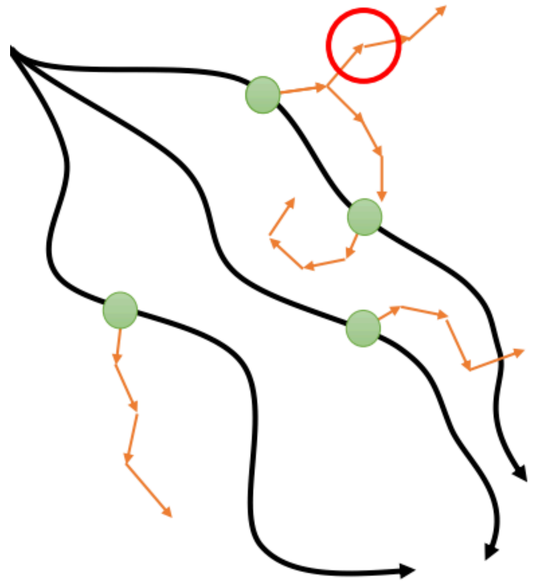
- Importance sampling for policy gradient methods.
- Model-based policy optimization.
- Action-value offline RL methods.
- Decision transformers.

Policy Gradients via Importance Sampling

- Recall policy gradient learning:
  - $\nabla_\theta J(\pi_\theta) = \mathbf{E}[q_\pi(s_t, a_t) \nabla_\theta \log \pi_\theta(a_t \mid s_t) \mid s_t \sim d_{\pi_\theta}, a_t \sim \pi_\theta]$
- Gradient is an expectation w.r.t. on-policy distribution.
  - Approximation with $\mathscr{D}$ provides a biased estimate of the gradient.
- One solution: correct with importance sampling.
  - $\nabla_\theta J(\pi_\theta) = \mathbf{E}[\frac{\pi_\theta(a_t \mid s_t)}{\pi_\beta(a_t \mid s_t)} q_\pi(s_t, a_t) \nabla_\theta \log \pi_\theta(a_t \mid s_t) \mid s_t \sim d_{\pi_\theta}, a_t \sim \pi_\theta].$
- Limitations:
  - Requires π_β is known or first estimated, e.g., with maximum likelihood $\hat{\pi}_\beta = \arg\max_\pi \sum_{i=1}^{m} \log \pi(a_t \mid s_t).$
  - High variance unless $\pi_\beta \approx \pi_\theta$.

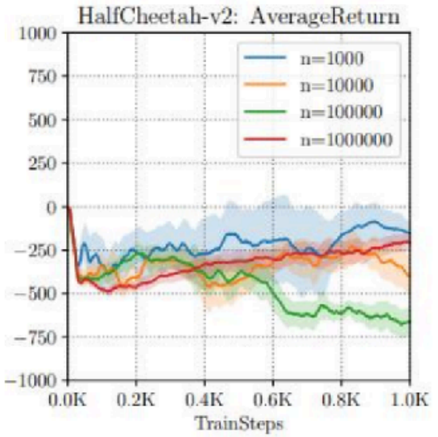
Model-based Offline RL

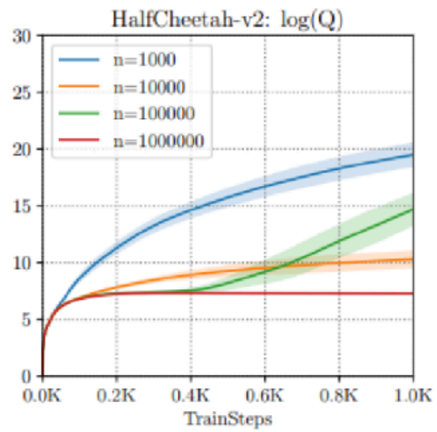
- Use $\mathscr{D}$ to build a simulator of the target MDP.
  - Use $\mathscr{D}$ to learn transition dynamics, p.
  - Learn $\pi^\star$ in the simulator.
- Limitations
  - Learning accurate models from scratch is hard.
  - What should the model predict when an action has not been observed?
- One solution: penalize the policy learned in simulation to avoid out of distribution actions, e.g., with a reward penalty $\tilde{r}(s,a) = r(s,a) - \lambda u(s,a)$.



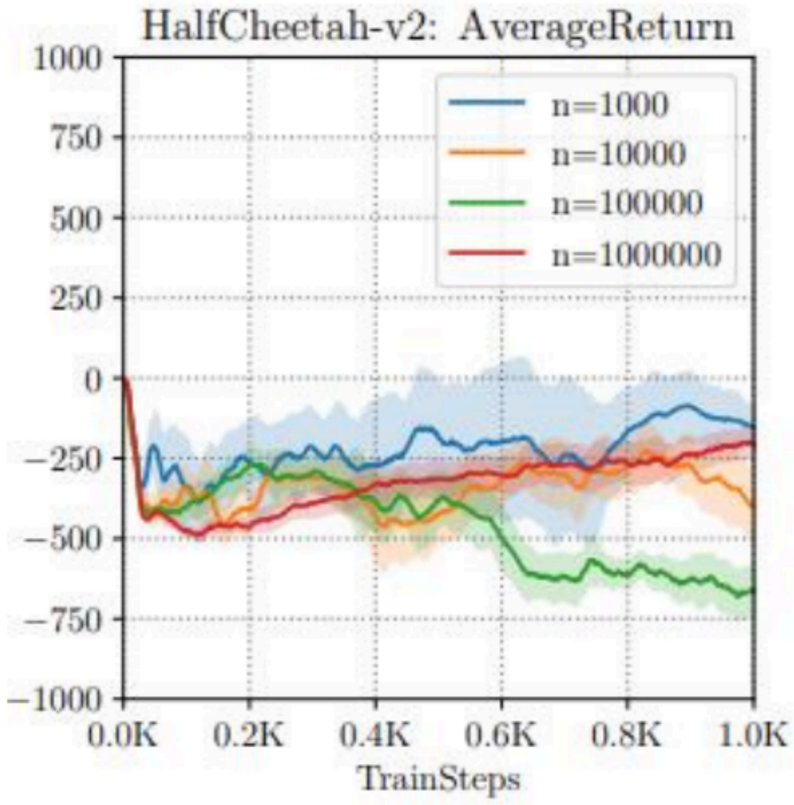
Action-value Based Offline RL

- Q-learning is already an off-policy algorithm.
  What if we just apply it directly to $\mathscr{D}$?
  - $q_{k+1}(s_i, a_i) \leftarrow q_k(s_i, a_i) + \alpha(r_i + \gamma \max_{a'} q_k(s_i', a') - q_k(s_i, a_i))$
  - $\max_{a'} q_k(s_i', a')$ might over-estimate value if a' is not in the data.





Overly Optimistic Q-Learning



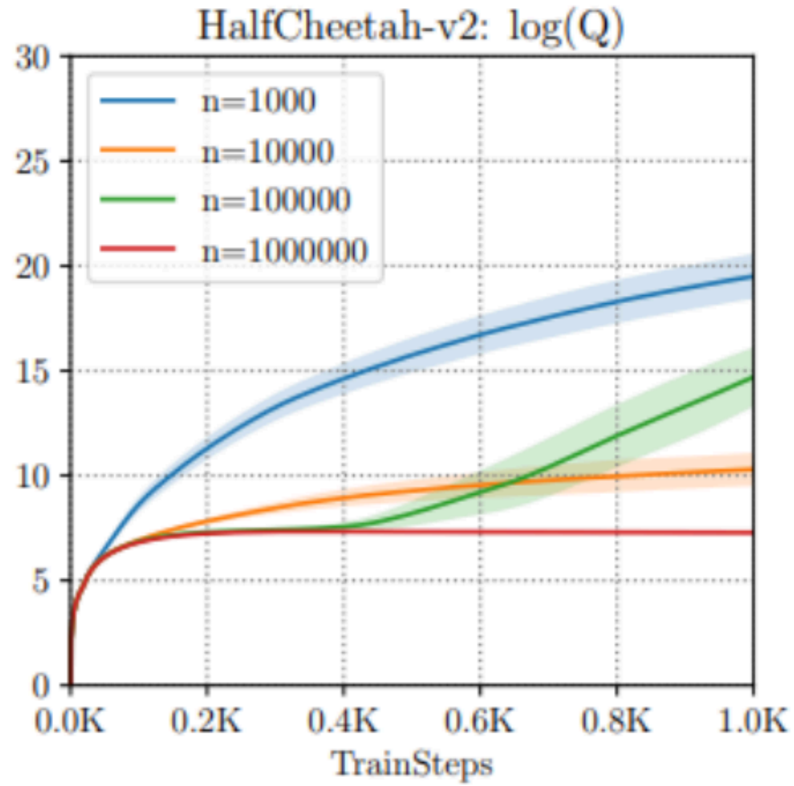


Image Credit: Conservative Q-Learning for Offline Reinforcement Learning. Kumar et al. 2019.

Constrained Policy Iteration

- Possible fix to over-estimation: keep current policy close to π_β.
  - Close in terms of KL-divergence [1] or maximal mean discrepancy [2].
- $\pi_{k+1} = \arg\max_{\pi} \mathbf{E}[q_{\pi_k}(s, a) \,|\, s \sim \mathscr{D}, a \sim \pi]$ such that $d(\pi_\beta || \pi) \leq \epsilon$.
- Intuition: make local improvement on top of π_β.
- Limitations: may not know π_β; unclear how to estimate it.
- One solution: use an implicit constraint.

[1] Behavior Regularized Offline Reinforcement Learning. Wu et al. 2019.
[2] Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction. Kumar et al. 2019.

Conservative Q-Learning

- Be pessimistic with out-of-distribution action-values.

$$\mathcal{L}_{CQL} = (Q(s,a) - (r + \gamma \mathbf{E}_{\pi}[Q(s',a')]))^2 - \alpha \mathbf{E}_{(s,a)\sim\mathcal{D}}[Q(s,a)] + \max_{\mu} \mathbf{E}_{s\sim\mathcal{D},a\sim\mu(a|s)}[Q(s,a)]$$

Expected SARSA
$Q \to q_{\pi}$

In-distribution bonus
$Q \to \infty$

OOD penalty
$Q \to 0$

- Make π greedy w.r.t. Q and repeat.

- Limitations: when to stop training to avoid over-fitting? We lack offline RL workflows as we have with supervised learning.

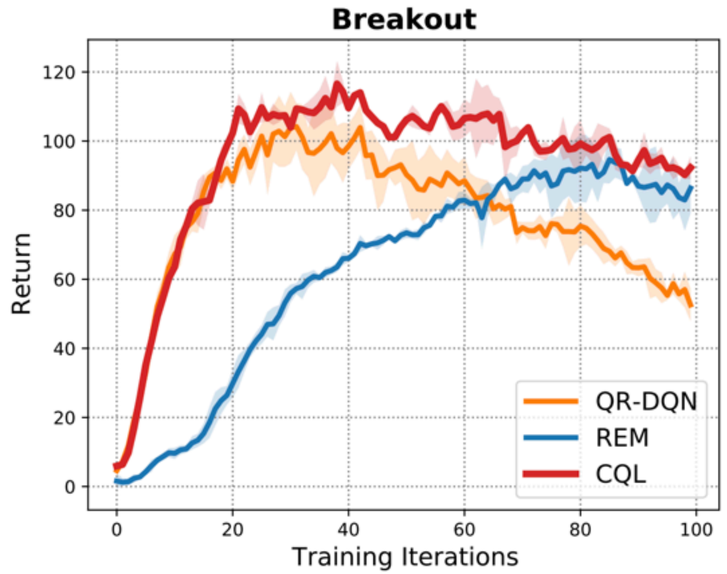


Image Credit: Conservative Q-Learning for Offline Reinforcement Learning. Kumar et al. 2019.

Advanced Challenges

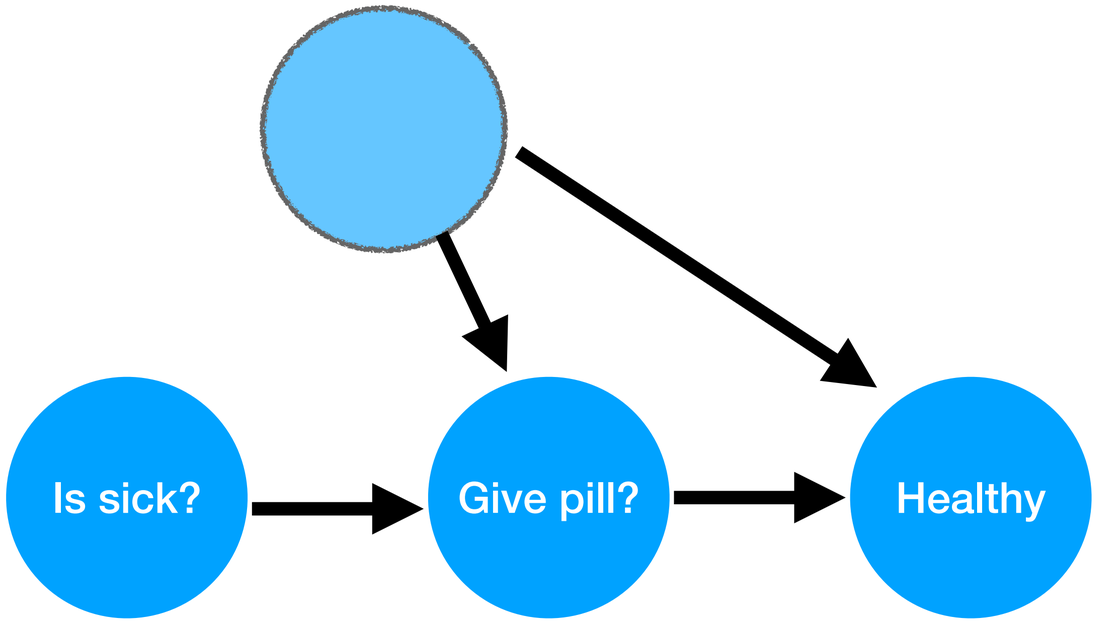
- Non-stationarity: offline data was collected in the past and the target MDP may have changed.
- Offline data may lack rewards or actions.
  - Example: videos of a task show you **what** happened but not **how** done.
- Partial observability:
  - Markov assumption might be violated.
  - Unobserved confounders.

Unobserved Confounders

- So far we have assumed the data was generated by $\pi_\beta(a \mid s)$ meaning that the behavior policy based its action on the state s that we observe in the data.
- What if the behavior policy had access to information not recorded in the data?
- Example:
  - We have medical data that records a patient's vital signs, a treatment prescribed by a doctor, and whether the patient recovered or not.
  - Doctor observes — but does not record — the wealth of the patient.

Unobserved Confounders

Data Generating Process



π_β: **if rich and sick, give pill else don't.**

The Data

{sick, pill, healthy}
{sick, no pill, not healthy}
{not sick, no pill, healthy}
{not sick, no pill, healthy}

Assume wealth leads to recovery (e.g., better diet) and affects doctor's decision.

Even if the pill is useless, an online RL algorithm will conclude that it is beneficial!

Off-Policy Evaluation

- In offline RL, the learned policy does not interact with the real world until deployment time.
  - How do we know that a learned policy will perform well?
  - How do we select hyper-parameters for RL algorithms?
- Answer: use $\mathscr{D}$ to estimate $J(\pi)$ for learned policy π.

What would the expected return be had we ran π instead of π_β?

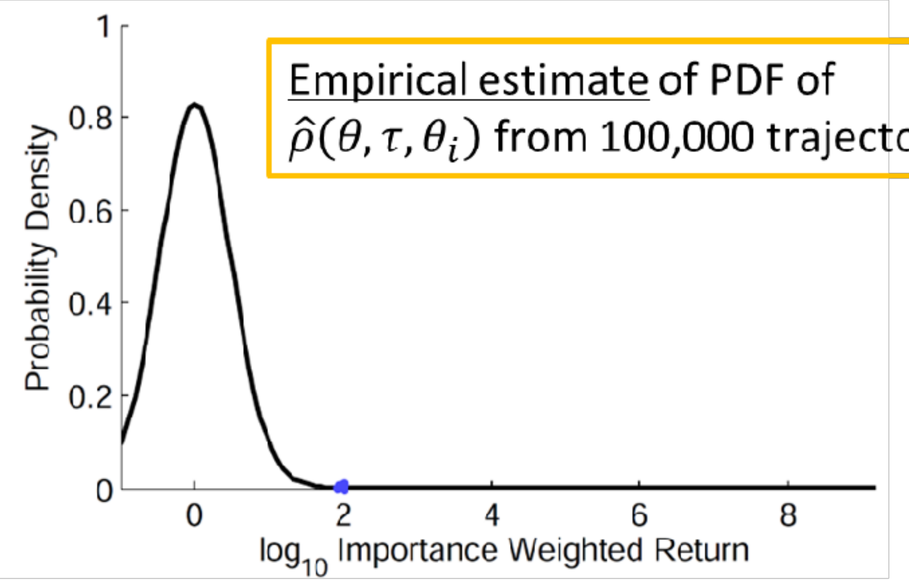
Importance Sampling Policy Evaluation

- Assume $\mathscr{D}$ consists of full episodes, $\mathscr{D} = \{(S_0, A_0, R_0, S_1, \ldots, S_T, A_T, R_t)\}$.
- If $\mathscr{D}$ had been generated by target policy π then $\frac{1}{m}\sum_{i=1}^{m}\sum_{t=0}^{T}\gamma^t R_t^i$ is an unbiased estimator of $J(\pi)$.
- Since $\mathscr{D}$ was generated by π_β, we instead use importance sampling to adjust for distribution shift:
  - $\widehat{J}(\pi) \approx \frac{1}{m}\sum_{i=1}^{m}\rho_i\sum_{t=0}^{T}\gamma^t R_t^i$ $\qquad \rho_i = \prod_{t=0}^{T}\frac{\pi(A_t^i \mid S_t^i)}{\pi_\beta(A_t^i \mid S_t^i)}$
- Limitations: high variance; requires π_β is known or estimated.
- Can be improved with different variance reduction techniques: weighted IS, control variates.

Approach – generating unbiased estimates of $\rho(\theta)$

- Unbiased estimate $\hat{\rho}(\theta, \tau, \theta_i)$ generated using importance sampling

$$\hat{\rho}(\theta, \tau, \theta_i) = R(\tau)\frac{\Pr(\tau|\theta)}{\Pr(\tau|\theta_i)} := \underbrace{R(\tau)}_{\text{return}} \underbrace{\prod_{t=1}^{T} \frac{\pi(a_t|s_t, \theta)}{\pi(a_t|s_t, \theta_i)}}_{\text{importance weight}}$$



- $\hat{\rho}(\theta, \tau, \theta_i)$ is bounded from below by zero
  - Since returns are normalized to $[0, 1]$
- Upper bound:
  - Probability of selection of a specific action could be low under behavior policy and high under evaluation policy - makes the importance weighted return be large
  - $\hat{\rho}(\theta, \tau, \theta_i)$ has expected value in [0, 1] and has a long tail (large upper bound)
- Hence need to account for large range and high-variance to produce a tight bound on $\rho(\theta)$

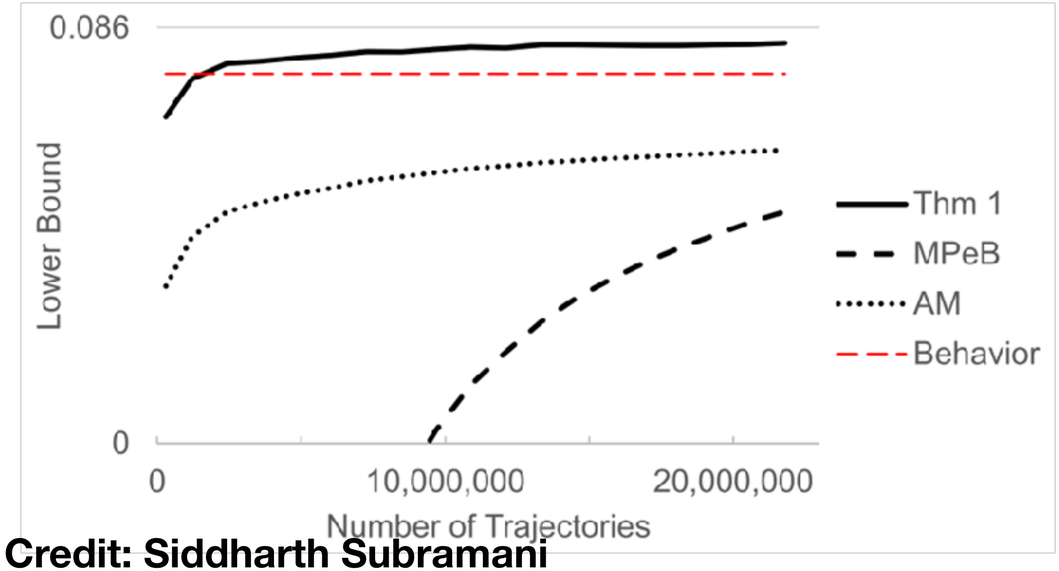
Slide Credit: Siddharth Subramani

Experiments and results

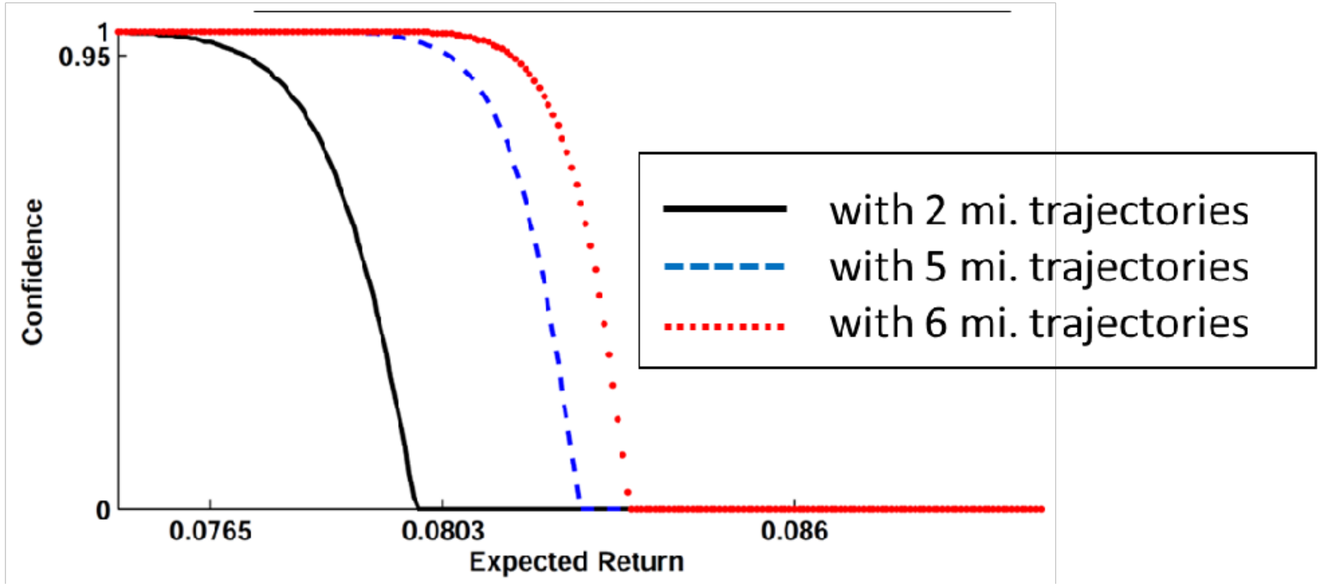
Targeting digital advertisement

- Ads shown on a webpage is based on known features of a user
  - Problem that attempts to maximize the probability of user clicking an ad
  - Sparse reward problem – returns have high variance since most trajectories provide none to less feedback
- This paper uses data from Adobe simulator
  - 31 features representing each user, $+1$ reward when ad is clicked, 0 when ad is overlooked, $T = 20, \gamma = 1$

95% confidence lower bounds



Confidence that $\rho(\theta)$ is the lower bound



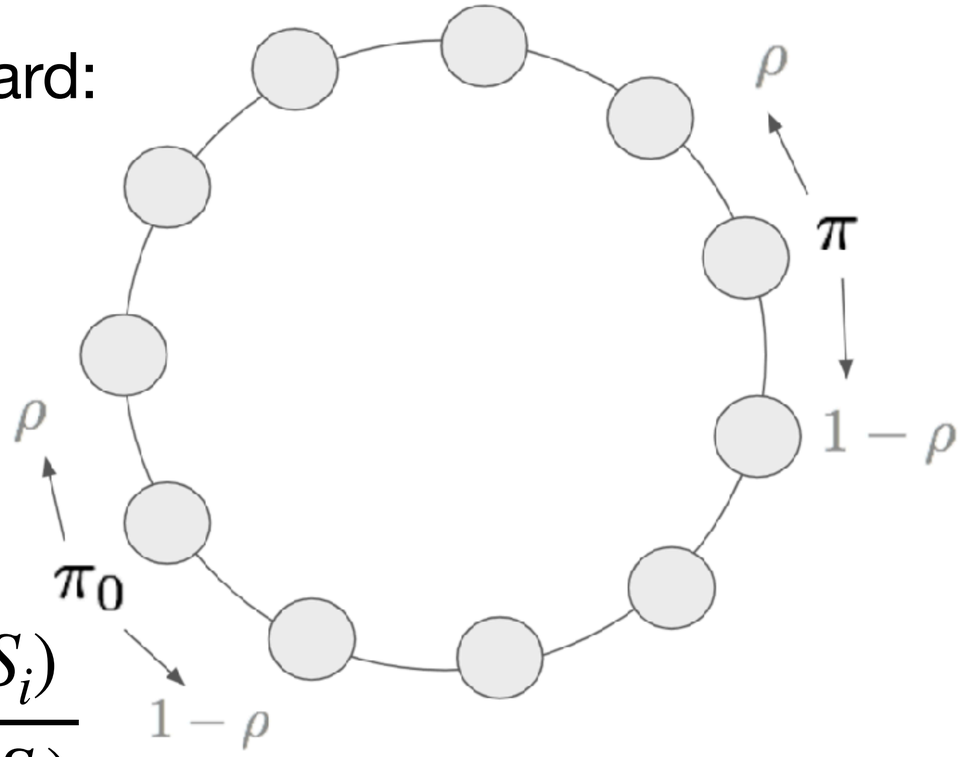
Slide Credit: Siddharth Subramani

Importance Sampling Policy Evaluation

- Importance sampling has variance that is exponential in the length of episodes.
- Alternatively, consider estimating average reward:
  - $J(\pi) = \frac{1}{1-\gamma}\mathbf{E}[R_t \mid S_t \sim d_\pi, A_t \sim \pi]$
  - $J(\pi) \approx \frac{1}{m}\sum_{i=1}^{m} w_i R_i$

Must be estimated from $\mathscr{D}$.
Many ways to do this.

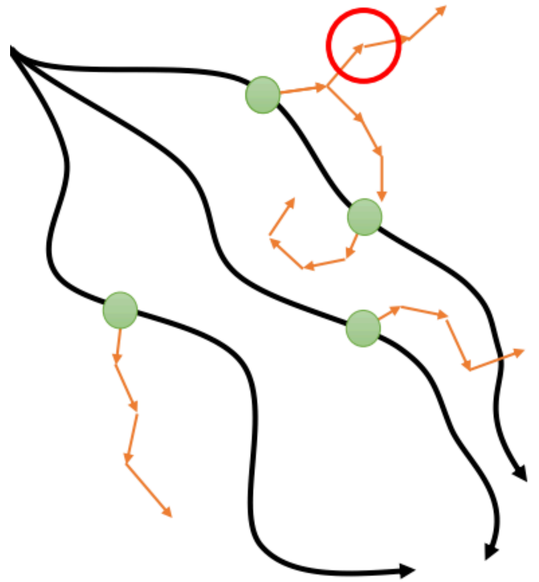
$$w_i = \frac{d_\pi(S_i, A_i)}{d_\beta(S_i, A_i)} = \frac{d_\pi(S_i)\pi(A_i \mid S_i)}{d_\beta(S_i)\pi_\beta(A_i \mid S_i)}$$



Breaking the Curse of Horizon: Infinite-Horizon Off-Policy Estimation. Liu et al. 2018

Model-based Policy Evaluation

- Use $\mathscr{D}$ to build a simulator of the target MDP.
  - Use $\mathscr{D}$ to learn transition dynamics, p.
  - Evaluate in the simulator.
- Limitations
  - Learning accurate models from scratch is hard.
  - What should the model predict when an action has not been observed?



Fitted Q-Evaluation

- Write policy performance in terms of action-values:
  - $J(\pi) = \mathbf{E}[q_\pi(S, A) \,|\, S \sim d_0, A \sim \pi]$
- Estimate q_π with DQN-like variant of expected SARSA:
  - $$\mathscr{L}(Q_\theta) = \frac{1}{m} \sum_{i=1}^{m} \left(r_i + \gamma \sum_{a'} \pi(a' \,|\, s_i') Q_{\bar{\theta}}(s_i', a') - Q_\theta(s_i, a_i) \right)^2$$

Like DQN except use expectation w.r.t. π instead of max

Which OPE method to use?

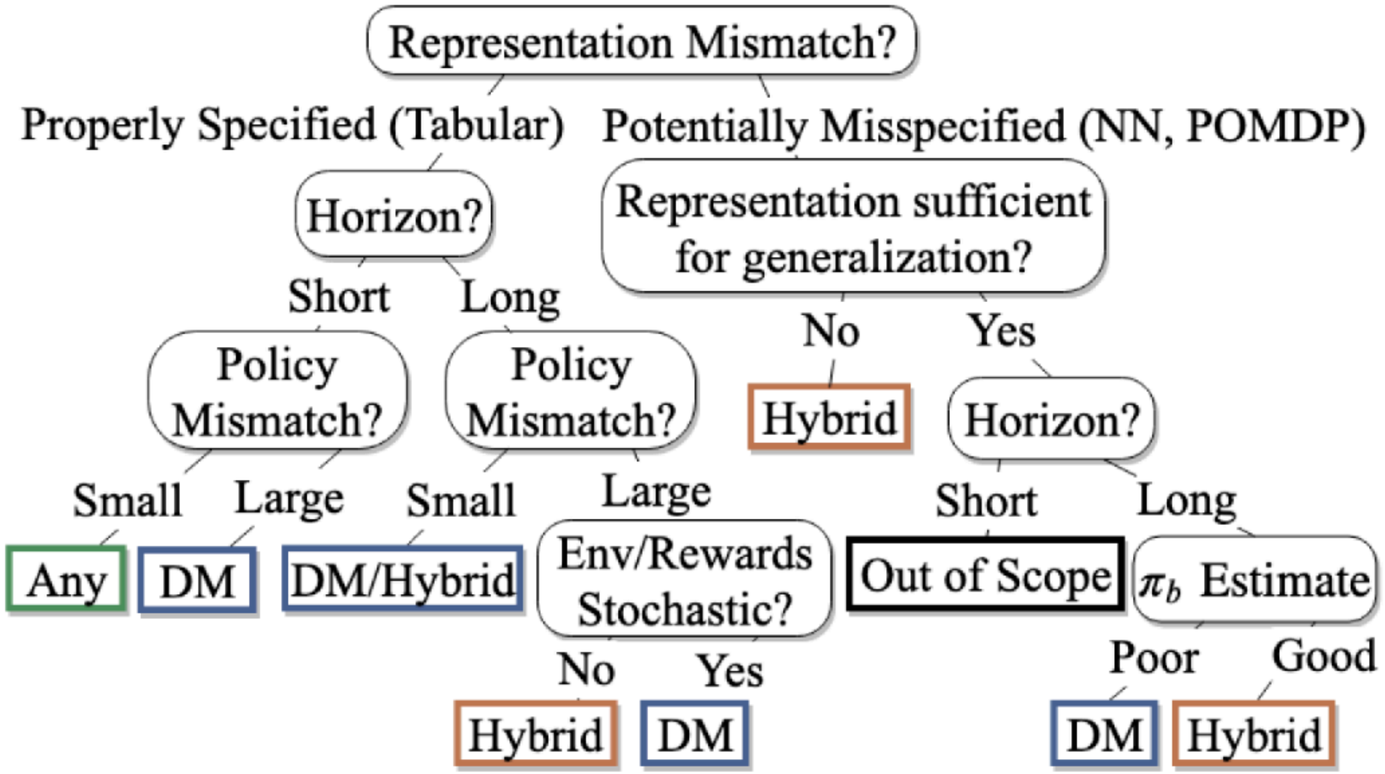


Figure 2: *General Guideline Decision Tree.*

Image Credit: Empirical Study of Off-Policy Policy Evaluation for Reinforcement Learning. Voloshin et al. 2021.

Summary

- Offline RL is RL with a static batch of data.
  - No exploration!
- Existing RL algorithms must be adapted for the offline setting:
  - Policy gradient methods may require importance sampling.
  - Model-based methods may require a pessimism assumption.
  - Q-learning-based methods also require pessimism.