Statistical limits of offline/batch RL

Yingru Li

The Chinese University of Hong Kong, Shenzhen, China

December 10, 2020

Outline

Introduction

- Motivation
- Offline RL Problems

Statistical limits

- Finite horizon episodic setting
- Infinite horizon discounted setting

Outline

Introduction

- Motivation
- Offline RL Problems

Statistical limits

- Finite horizon episodic setting
- Infinite horizon discounted setting

Current RL success paradigm

- RL algorithms can learn complex behaviors in simulation, where active (on-policy) data collection is straightforward.

Figure: Go and Game: 'good simulator ≈ infinite accessible data with almost no expense as long as the computation resources is provided'

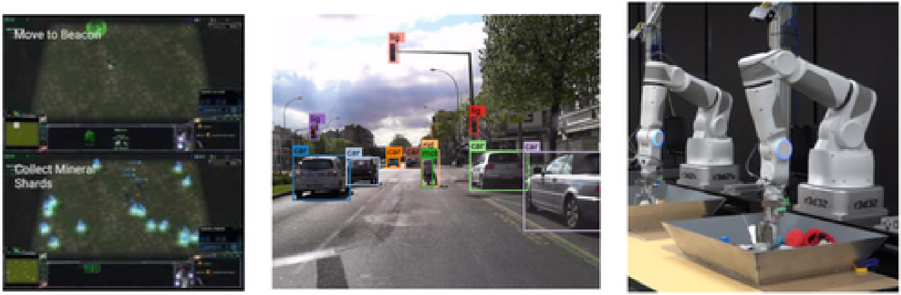
Motivation: the real-world applications

- In real-world applications, the performance is limited by the expense of active data collection.
  - Deploying a policy to collect new data is costly. (E.g. Recommendation systems, DiDi/Uber.)
  - Safety concern with updating/executing the policy online. (E.g. Robotic control, Healthcare applications, autonomous driving, communication networks.)
- Deploying a new policy may only be done at a **low frequency** after extensive testing and evaluation.
- Good news:
  - In some of these cases, the offline dataset are often very large, potentially encompassing years of logged experience. (**Our focus today**)
  - We can build good simulators based on some specific applications and try to transfer what we learn in simulators to real environments. (Sim2Real)

Motivation: the real-world applications

- In real-world applications, the performance is limited by the expense of active data collection.
  - Deploying a policy to collect new data is costly. (E.g. Recommendation systems, DiDi/Uber.)
  - Safety concern with updating/executing the policy online. (E.g. Robotic control, Healthcare applications, autonomous driving, communication networks.)
- Deploying a new policy may only be done at a **low frequency** after extensive testing and evaluation.
- Good news:
  - In some of these cases, the offline dataset are often very large, potentially encompassing years of logged experience. (**Our focus today**)
  - We can build good simulators based on some specific applications and try to transfer what we learn in simulators to real environments. (Sim2Real)

Offline Datasets



Starcraft Replays (1M)

Self-driving cars (1100h)

Robotic Grasping (1M)

Figure: We want to make use of these fixed and static offline datasets when doing RL as environment interaction is (often) costly and even dangerous.

Outline

Introduction

Motivation

Offline RL Problems

Statistical limits

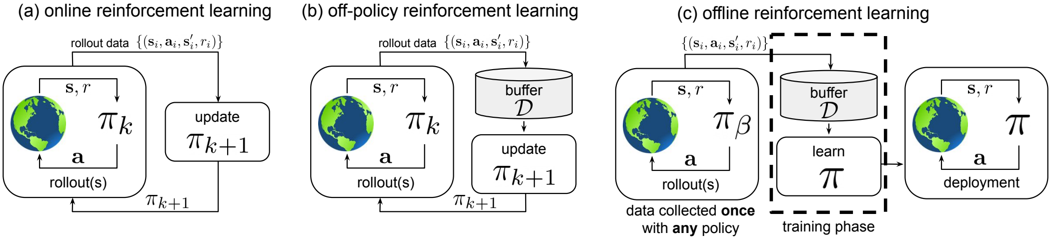
Finite horizon episodic setting

Infinite horizon discounted setting

Offline Reinforcement Learning

Fundamental Question:

How to effectively utilize offline datasets for future decisions while the agents are not able to interact with the environment to gather new data?



(a) online reinforcement learning

(b) off-policy reinforcement learning

(c) offline reinforcement learning

Figure: Pictorial illustration of classic online reinforcement learning (a), classic off-policy reinforcement learning (b), and offline reinforcement learning (c). In online reinforcement learning. (Figure from [Levine, Kumar, Tucker, and Fu, 2020])

Offline Reinforcement Learning

Fundamental Question:

How to effectively utilize offline datasets for future decisions while the agents are not able to interact with the environment to gather new data?

- Learning and generalizing by incorporating diverse historical experience without further trial and errors,
  - Not just imitating historical experience. (Introspective Intelligence)
- Better sample efficiency.

Offline RL Problematics (I)

Insufficient coverage and Distributional shift

- **Fixed under-explored offline dataset**: dataset without enough exploration often cannot cover enough states and actions.
- Even for **tabular setting**, there is no guarantee that the optimal policy can be found using the under-explored dataset.
  - Not possible to find optimal policy with little data coverage on the state-action region that optimal policy frequently visits.
- Problems with large or continuous state and action spaces require **function approximation** to generalize across states and actions.
  - Under-explored data will lead to erroneous generalization of the function for state-action pairs in under-explored region.

Offline RL Problematics (II)

Extrapolation error from distributional shift

- Problems with large or continuous state and action spaces require function approximation.
- Erroneous generalization/extrapolation error of the state-action value function (Q-value function) learned with function approximators leads to high bootstrapping error. [Kumar et al., 2019, Wu et al., 2019]

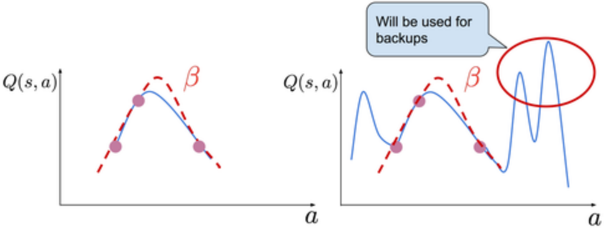


Figure: Incorrectly high Q-values for OOD actions may be used for backups, leading to accumulation of error.

Offline RL Problematics (III)

Boostrapping Error

- Suppose the offline dataset is collected by the behavior policy $\pi_\beta(a|s)$ (possibly multiple).
- For one transition tuple collected by behaviour policy $\pi_\beta(a|s)$ with policy induced state-action distribution $\beta(s,a)$:

$$(s,a,s') \sim \beta(s,a)P(s'|s,a)$$

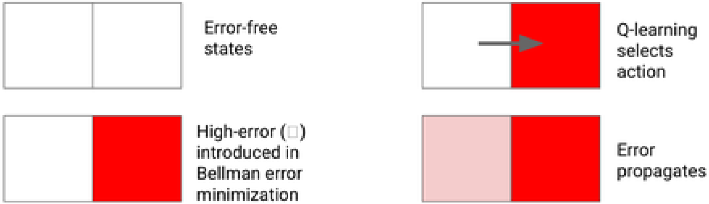
- Illustration via Q value iteration:

$$\underbrace{Q^{k+1}(s,a)}_{\text{Errors accumulated into } Q(s,a)} \leftarrow r(s,a) + \gamma \underbrace{\max_{a'} Q^k(s',a')}_{\text{usually query at unseen } a'}$$

  - $Q(s',a')$ for $s' \nsim \beta$: Out-of-distribution (OoD) state
  - $Q(s',a')$ for $s' \sim \beta$, a' far from $\pi_\beta(a'|s')$: OoD action.

Offline RL Problematics (IV)

Error Propagation



Error-free states

Q-learning selects action

High-error (□) introduced in Bellman error minimization

Error propagates

$$\epsilon^{k+1}(s,a) \leq \delta(s,a) + \gamma\epsilon^{k}(s',a')$$

Overall Error (Q-Q*)

Bellman Error

Propagated error

Offline RL Problematics Lead to Wrong Behavior Consequences

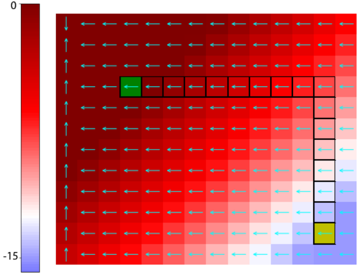


Figure: Learning goal-reaching policy from offline dataset $\mathcal{D}$. **Wrongly linear extrapolation!** (Figure from [Luo et al., 2019])

- Reward = -1 if not reaching the goal
- V^* = - minkovski distance to goal
- Learned (linear) value function
  - Correct within the support of offline dataset $\mathcal{D}$
  - Wrong outside the support
- Resulting wrong behavior induced from learned value
- Conclusions: Learning from $\mathcal{D}$ only guarantees accurate predictions on the offline data distribution
  - e.g. Q-learning with $\mathcal{D}$ results over-estimation outside the support of $\mathcal{D}$.

Outline

Introduction

- Motivation
- Offline RL Problems

Statistical limits

- Finite horizon episodic setting
- Infinite horizon discounted setting

Outline

Introduction

- Motivation
- Offline RL Problems

Statistical limits

- Finite horizon episodic setting
- Infinite horizon discounted setting

Statistical limits analysis setup in the episodic RL setting

- MDP $M = (\mathcal{S}, \mathcal{A}, P, R, H)$
  - State space $\mathcal{S}$, action space $\mathcal{A}$, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition operator, $R : \mathcal{S} \times \mathcal{A} \to \Delta(\mathbb{R})$ is the reward distribution. $H \in \mathbb{Z}_+$ is the planning horizon
- For simplicity, we assume a fixed initial state $s_1 \in \mathcal{S}$.
- A (stochastic) policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ chooses an action a randomly based on the current state s.
- The policy π induces a (random) trajectory $s_1, a_1, r_1, s_2, a_2, r_2, \ldots, s_H, a_H, r_H$, where $a_1 \sim \pi_1(s_1)\ r_1 \sim R(s_1, a_1), s_2 \sim P(s_1, a_1), a_2 \sim \pi_2(s_2)$, etc.
- To streamline our analysis, for each $h \in [H]$, we use $\mathcal{S}_h \subseteq \mathcal{S}$ to denote the set of states at level h, and we assume $\mathcal{S}_h$ do not intersect with each other.
- We assume, almost surely, that $r_h \in [-1, 1]$ for all $h \in [H]$.

Statistical limits analysis setup in the episodic RL setting

- **Value Functions**: Given a policy $\pi, h \in [H]$ and $(s, a) \in \mathcal{S}_h \times \mathcal{A}$, the Q -function and value function are defined as:

$$Q_h^\pi(s, a) = \mathbb{E}\left[\sum_{h'=h}^{H} r_{h'} \mid s_h = s, a_h = a, \pi\right], \quad V_h^\pi(s) = \mathbb{E}\left[\sum_{h'=h}^{H} r_{h'} \mid s_h = s, \pi\right]$$

- For a policy π, we define $V^\pi = V_1^\pi(s_1)$ to be the value of π from the fixed initial state s_1.

Linear Function Approximation and Realizability Assumption

- **Linear Function Approximation.** A feature extractor $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$
  - either hand-crafted feature extractor or a pre-trained neural network that transforms a state-action pair to a d -dimensional embedding
  - and the Q-functions can be predicted by linear functions of the features.
- **Assumption 1**: Linear Q^π realizability (Realizable Linear Function Approximation).
- For every policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, there exists $\theta_1^\pi, \ldots \theta_H^\pi \in \mathbb{R}^d$ such that for all $(s, a) \in \mathcal{S} \times \mathcal{A}$ and $h \in [H]$,

$$Q_h^\pi(s, a) = \left(\theta_h^\pi\right)^\top \phi(s, a)$$

Offline RL setting for statistical limits analysis

- In Offline RL setting,
  - the agent does **not** have direct access to the MDP/Environment
  - and instead is given access to **data distributions** $\{\mu_h\}_{h=1}^{H}$ where for each $h \in [H], \mu_h \in \Delta\left(\mathcal{S}_h \times \mathcal{A}\right)$ for analysis.
- The inputs of the agent are H datasets $\{D_h\}_{h=1}^{H}$,
- and for each $h \in [H], D_h$ consists i.i.d. samples of the form $(s, a, r, s') \in \mathcal{S}_h \times \mathcal{A} \times \mathbb{R} \times \mathcal{S}_{h+1}$ tuples, where $(s, a) \sim \mu_h, r \sim r(s, a), s' \sim P(s, a)$.

Offline evaluation problem

- Here we focus on the offline policy evaluation problem with linear function approximation:
- Given a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ and a feature extractor $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$,
- **Goal**: output an accurate estimate of the value of π (i.e., V^π) approximately, using the collected datasets $\{D_h\}_{h=1}^H$, with as few samples as possible.

Other notations will be used later

- For a vector $x \in \mathbb{R}^d$, we use $\|x\|_2$ to denote its ℓ_2 norm.
- For a positive semidefinite matrix A, we use $\|A\|_2$ to denote its operator norm, and $\sigma_{\min}(A)$ to denote its smallest eigenvalue.
- For two positive semidefinite matrices A and B, we write $A \succeq B$ to denote the Löwner partial ordering of matrices, i.e, $A \succeq B$ if and only if $A - B$ is positive semidefinite.
- For a policy $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$, we use μ_h^π to denote the marginal distribution of s_h under π, i.e., $\mu_h^\pi(s) = \Pr\left[s_h = s \mid \pi\right]$.
- For a vector $x \in \mathbb{R}^d$ and a positive semidefinite matrix $A \in \mathbb{R}^{d \times d}$, we use $\|x\|_A$ to denote $\sqrt{x^\top A x}$.

Sufficient feature coverage assumption

- **Assumption 2 (Feature Coverage)**.
- For all $(s, a) \in \mathcal{S} \times \mathcal{A}$, assume our feature map is bounded such that $\|\phi(s, a)\|_2 \leq 1$.
- Furthermore, suppose for each $h \in [H]$, the data distributions μ_h satisfy the following minimum eigenvalue condition:

$$\sigma_{\min}\left(\mathbb{E}_{(s,a)\sim\mu_h}\left[\phi(s, a)\phi(s, a)^\top\right]\right) = 1/d$$

- **Note** that $1/d$ is the largest possible minimum eigenvalue due to that, for any data distribution $\widetilde{\mu}_h$, $\sigma_{\min}\left(\mathbb{E}_{(s,a)\sim\widetilde{\mu}_h}\left[\phi(s, a)\phi(s, a)^\top\right]\right) \leq \frac{1}{d}$ by the fact $\|\phi(s, a)\|_2 \leq 1$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$ and the trace argument.

Remark: when the horizon $H = 1$

- **Remark**: Clearly, for the case where $H = 1$, the realizability assumption (Assumption 1), and feature coverage assumption (Assumption 2) imply that the ordinary least squares estimator will accurately estimate θ_1[Hsu, Kakade, and Zhang, 2014].
- Main result in this paper shows that these assumptions are not sufficient for offline policy evaluation for long horizon problems.

Worst-case Lower bound for Offline Policy Evaluation problem

Theorem 1.

*Suppose **Assumption 2** holds. Fix an algorithm that takes as input both a policy and a feature mapping. There exists a (deterministic) MDP satisfying **Assumption 1**, such that for any policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, the algorithm requires $\Omega\left((d/2)^H\right)$ samples to output the value of π up to constant additive approximation error with probability at least 0.9.*

Remarks on the Lower bound

- **Remark 1 (The sparse reward case).** As stated, the theorem uses a deterministic MDP (with stochastic rewards). See Appendix A for another hard case where the transition is stochastic and the reward is deterministic and sparse (only occurring at two states at $h = H$).

Remarks on the Lower bound

- **Remark 2 (Least-Squares Policy Evaluation (LSPE) has exponential variance).**
- Most näve algorithm here would be LSPE,
  - i.e., using ordinary least squares (OLS) to estimate θ^π, starting at level $h = H$
  - and then proceeding backwards to level $h = 1$, using the plug-in estimator from the previous level.
- Here, LSPE will provide an unbiased estimate (provided the feature covariance matrices are full rank, which will occur with high probability).
- Interestingly, as a direct corollary, the above theorem implies that LSPE has exponential variance in H.
- Later we will have a more detailed discussion on LSPE. More generally, our theorem implies that there is no estimator that can avoid such exponential dependence in the offline setting.

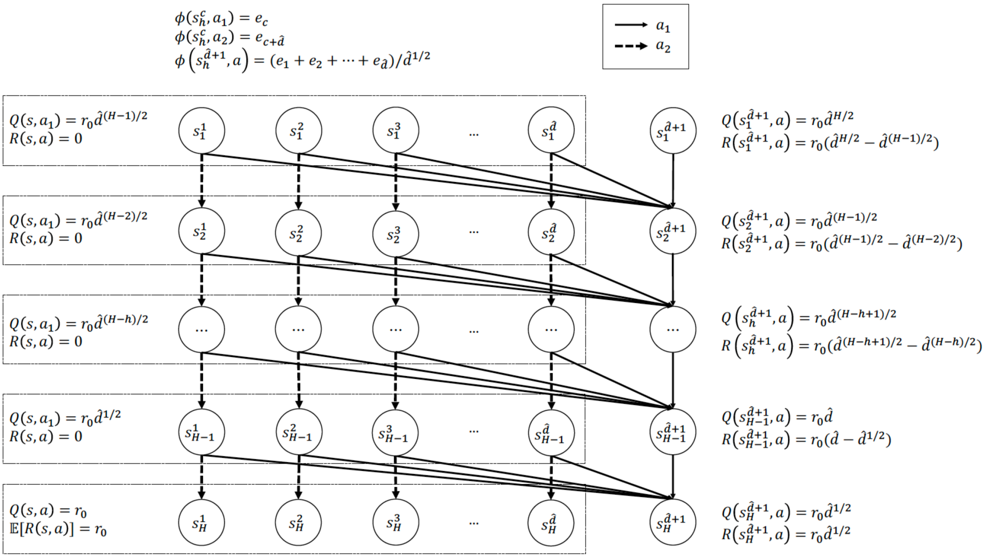
Remarks on the Lower bound

- **Remark 3 (Least-Squares Value Iteration (LSVI) versus Least-Squares Policy Iteration (LSPI)).**
- The most naive algorithm here would be LSVI,
  - i.e., using ordinary least squares (OLS) to estimate θ^*, starting at level $h = H$
  - and then proceeding backwards to level $h = 1$, using the plug-in estimator from the previous level and the bellman operator.
- As a corollary, the above theorem implies that LSVI will require an exponential number of samples to find a near-optimal policy.
- On the other hand, if the regression targets are collected by using rollouts (i.e. on-policy sampling) as in LSPI [Lagoudakis and Parr, 2003], then a polynomial number of samples suffice. See Section D in [Du et al., 2020] for an analysis.
- Therefore, Theorem 4.1 implies an exponential separation on the sample complexity between LSVI and LSPI. Of course, LSPI requires adaptive data samples (why?) and thus does not work in the offline setting.

Hard instance construction

- Feature dimension d assumed to be even. **Denote $\hat{d} := d/2$ for convenience.**
- **State Space, Action Space and Transition Operator.**
- The action space $\mathcal{A} = \{a_1, a_2\}$.
- For each $h \in [H], \mathcal{S}_h$ contains $\hat{d} + 1$ states $s_h^1, s_h^2, \ldots, s_h^{\hat{d}}$ and $s_h^{\hat{d}+1}$.
- For each $h \in [H-1]$, for each $c \in \{1, 2, \ldots, \hat{d}+1\}$, we have

$$P\left(s_h^c, a\right) = \begin{cases} s_{h+1}^{\hat{d}+1} & a = a_1 \\ s_{h+1}^c & a = a_2 \end{cases}$$



a_1
a_2
s_1^1 s_1^2 s_1^3 ... $s_1^{\hat{d}}$ $s_1^{\hat{d}+1}$
s_2^1 s_2^2 s_2^3 ... $s_2^{\hat{d}}$ $s_2^{\hat{d}+1}$
... ...
s_{H-1}^1 s_{H-1}^2 s_{H-1}^3 ... $s_{H-1}^{\hat{d}}$ $s_{H-1}^{\hat{d}+1}$
s_H^1 s_H^2 s_H^3 ... $s_H^{\hat{d}}$ $s_H^{\hat{d}+1}$

$\phi(s_h^c, a_1) = e_c$
$\phi(s_h^c, a_2) = e_{c+\hat{d}}$
$\phi\left(s_h^{\hat{d}+1}, a\right) = (e_1 + e_2 + \cdots + e_{\hat{d}})/\hat{d}^{1/2}$

$Q(s, a_1) = r_0 \hat{d}^{(H-1)/2}$
$R(s, a) = 0$

$Q(s_1^{\hat{d}+1}, a) = r_0 \hat{d}^{H/2}$
$R(s_1^{\hat{d}+1}, a) = r_0(\hat{d}^{H/2} - \hat{d}^{(H-1)/2})$

$Q(s, a_1) = r_0 \hat{d}^{(H-2)/2}$
$R(s, a) = 0$

$Q(s_2^{\hat{d}+1}, a) = r_0 \hat{d}^{(H-1)/2}$
$R(s_2^{\hat{d}+1}, a) = r_0(\hat{d}^{(H-1)/2} - \hat{d}^{(H-2)/2})$

$Q(s, a_1) = r_0 \hat{d}^{(H-h)/2}$
$R(s, a) = 0$

$Q\left(s_h^{\hat{d}+1}, a\right) = r_0 \hat{d}^{(H-h+1)/2}$
$R\left(s_h^{\hat{d}+1}, a\right) = r_0(\hat{d}^{(H-h+1)/2} - \hat{d}^{(H-h)/2})$

$Q(s, a_1) = r_0 \hat{d}^{1/2}$
$R(s, a) = 0$

$Q(s_{H-1}^{\hat{d}+1}, a) = r_0 \hat{d}$
$R(s_{H-1}^{\hat{d}+1}, a) = r_0(\hat{d} - \hat{d}^{1/2})$

$Q(s, a) = r_0$
$\mathbb{E}[R(s, a)] = r_0$

$Q(s_H^{\hat{d}+1}, a) = r_0 \hat{d}^{1/2}$
$R(s_H^{\hat{d}+1}, a) = r_0 \hat{d}^{1/2}$

Hard instance construction

- **Reward Distributions.** Let $0 \leq r_0 \leq \hat{d}^{-H/2}$ be a parameter to be determined.
- For each $(h, c) \in [H-1] \times [\hat{d}]$ and $a \in \mathcal{A}$, we set $R\left(s_h^c, a\right) = 0$
- and for $c = \hat{d} + 1$, we set

$$R\left(s_h^{\hat{d}+1}, a\right) = r_0 \cdot \left(\hat{d}^{(H-(h-1))/2} - \hat{d}^{(H-h)/2}\right).$$

- For the last level H, for each $c \in [\hat{d}]$ and $a \in \mathcal{A}$, we set

$$R\left(s_H^c, a\right) = \begin{cases} 1 & \text{with probability } \left(1+r_0\right)/2 \\ -1 & \text{with probability } \left(1-r_0\right)/2 \end{cases}$$

so that $\mathbb{E}\left[R\left(s_H^c, a\right)\right] = r_0$. Moreover, for all actions $a \in \mathcal{A}$, $R\left(s_H^{\hat{d}+1}, a\right) = r_0 \cdot \hat{d}^{1/2}$

Hard instance construction

- **Feature Mapping.**
- Let $e_1, e_2, \ldots, e_d$ be a set of orthonormal vectors in $\mathbb{R}^d$. Here, one possible choice is to set $e_1, e_2, \ldots, e_d$ to be the standard basis vectors.
- For each $(h, c) \in [H] \times [\hat{d}]$, we set $\phi\left(s_h^c, a_1\right) = e_c, \phi\left(s_h^c, a_2\right) = e_{c+\hat{d}}$,
- and for $c = \hat{d} + 1$, set

$$\phi\left(s_h^{\hat{d}+1}, a\right) = \frac{1}{\hat{d}^{1/2}} \sum_{c \in [\hat{d}]} e_c$$

for all $a \in \mathcal{A}$

Verifying Realizability Assumption

Lemma 2.

For every policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, *for each* $h \in [H]$, *for all* $(s, a) \in \mathcal{S}_h \times \mathcal{A}$, *we have* $Q_h^\pi(s, a) = \left(\theta_h^\pi\right)^\top \phi(s, a)$ *for some* $\theta_h^\pi \in \mathbb{R}^d$

- Proof. We first verify Q^π is linear for the first $H - 1$ levels. For each $(h, c) \in [H - 1] \times [\hat{d}]$, we have

$$Q_h^\pi\left(s_h^c, a_1\right) = R\left(s_h^c, a_1\right) + R\left(s_{h+1}^{\hat{d}+1}, a_1\right) + R\left(s_{h+2}^{\hat{d}+1}, a_1\right) + \ldots + R\left(s_H^{\hat{d}+1}, a_1\right) = r_0 \cdot \hat{d}^{(H-h)/2}$$

- Moreover, for all $a \in \mathcal{A}$,

$$Q_h^\pi\left(s_h^{\hat{d}+1}, a\right) = R\left(s_h^{\hat{d}+1}, a\right) + R\left(s_{h+1}^{\hat{d}+1}, a_1\right) + R\left(s_{h+2}^{\hat{d}+1}, a_1\right) + \ldots + R\left(s_H^{\hat{d}+1}, a_1\right) = r_0 \cdot \hat{d}^{(H-h+1)/2}$$

Verifying Realizability Assumption (Cont.)

- Therefore, if we define

$$\theta_h^\pi = \sum_{c=1}^{\hat{d}} r_0 \cdot \hat{d}^{(H-h)/2} \cdot e_c + \sum_{c=1}^{\hat{d}} Q_h^\pi \left(s_h^c, a_2\right) \cdot e_{c+\hat{d}}$$

  then $Q_h^\pi(s,a) = \left(\theta_h^\pi\right)^\top \phi(s,a)$ for all $(s,a) \in \mathcal{S}_h \times \mathcal{A}$
- Now we verify that the Q-function is linear for the last level.
- Clearly, for all $c \in [\hat{d}]$ and $a \in \mathcal{A}, Q_H^\pi\left(s_H^c, a\right) = r_0$ and $Q_H^\pi\left(s_H^{\hat{d}+1}, a\right) = r_0 \cdot \sqrt{\hat{d}}$.
- Thus by defining $\theta_H^\pi = \sum_{c=1}^{d} r_0 \cdot e_c$, we have $Q_H^\pi(s,a) = \left(\theta_H^\pi\right)^\top \phi(s,a)$ for all $(s,a) \in \mathcal{S}_H \times \mathcal{A}$

Verifying the Feature Converage Assumption

- **The Data Distributions.** For each level $h \in [H]$, the data distribution μ_h is a **uniform distribution** over $\left\{ \left(s_h^1, a_1\right), \left(s_h^1, a_2\right), \left(s_h^2, a_1\right), \left(s_h^2, a_2\right), \ldots, \left(s_h^{\hat{d}}, a_1\right), \left(s_h^{\hat{d}}, a_2\right) \right\}$.
- Notice that $\left(s_h^{\hat{d}+1}, a\right)$ is not in the support of μ_h for all $a \in \mathcal{A}$.
- It can be seen that,

$$\mathbb{E}_{(s,a)\sim\mu_h}\left[\phi(s,a)\phi(s,a)^\top\right] = \frac{1}{d}\sum_{c=1}^{d} e_c e_c^\top = \frac{1}{d} I$$

Lower bound

- We show that it is information-theoretically hard for any algorithm to distinguish the case $r_0 = 0$ and $r_0 = \hat{d}^{-H/2}$.
- We fix the initial state to be $s_1^{\hat{d}+1}$, and consider any policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$.

a. When $r_0 = 0$, all reward values will be zero, and thus $V^\pi(s_1^{\hat{d}+1}) = 0$

b. When $r_0 = \hat{d}^{-H/2}$, the value of π would be $V^\pi(s_1^{\hat{d}+1}) = r_0 \cdot \hat{d}^{H/2} = 1$.

- Thus, if the algorithm approximates the value of the policy up to **an error of** $1/2$, then **it must distinguish the case that** $r_0 = 0$ **and** $r_0 = \hat{d}^{-H/2}$.

Lower bound

- For the case $r_0 = 0$ and $r_0 = \hat{d}^{-H/2}$, the data distributions $\{\mu_h\}_{h=1}^{H}$, the feature mapping $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, the policy π to be evaluated and the transition operator P are the same.
- Thus, in order to distinguish the case $r_0 = 0$ and $r_0 = \hat{d}^{-H/2}$, the only way is to query the reward distribution by using sampling taken from the data distributions.
- For all state-action pairs (s, a) in the support of the data distributions of the first $H - 1$ levels, the reward distributions will be identical. This is because for all $s \in \mathcal{S}_h \backslash \left\{ s_h^{\hat{d}+1} \right\}$ and $a \in \mathcal{A}$, we have $R(s, a) = 0$.
- For the case $r_0 = 0$ and $r_0 = \hat{d}^{-H/2}$, for all state-action pairs (s, a) in the support of the data distribution of the last level,

$$R(s, a) = \begin{cases} 1 & \text{with probability } \left(1 + r_0\right)/2 \\ -1 & \text{with probability } \left(1 - r_0\right)/2 \end{cases}$$

Lower bound

- Therefore, to distinguish the case that $r_0 = 0$ and $r_0 = \hat{d}^{-H/2}$, the agent needs to distinguish two reward distributions

$$r_1 = \begin{cases} 1 & \text{with probability } 1/2 \\ -1 & \text{with probability } 1/2 \end{cases}$$

  and

$$r_2 = \begin{cases} 1 & \text{with probability } \left(1 + \hat{d}^{-H/2}\right)/2 \\ -1 & \text{with probability } \left(1 - \hat{d}^{-H/2}\right)/2 \end{cases}$$

- It is well known that in order to distinguish r_1 and r_2 with probability at least 0.9, any algorithm requires $\Omega\left(\hat{d}^H\right)$ samples. See e.g. Lemma 5.1 in [Anthony and Bartlett, 2009]. See also [Chernoff, 1972, Mannor and Tsitsiklis, 2004].

Remark of the lower bound

- The key in our construction is the state $s_h^{\hat{d}+1}$ in each level, whose feature vector is defined to be $\sum_{c \in [\hat{d}]} e_c / \hat{d}^{1/2}$.
- In each level, $s_h^{\hat{d}+1}$ amplifies the Q-values by a $\hat{d}^{1/2}$ factor, due to the linearity of the Q-function.
- After all the H levels, the value will be amplified by a $\hat{d}^{H/2}$ factor.
- Since $s_h^{\hat{d}+1}$ is not in the support of the data distribution, the only way for the agent to estimate the value of the policy is to **estimate the expected reward value in the last level**.
- Our construction forces the estimation error of the last level to be amplified exponentially and thus implies an exponential lower bound.

Hardness reduction from policy optimization to policy evaluation

- Although we focus on offline policy evaluation in this work, our hardness result also holds for finding near-optimal policies under Assumption 1 in the offline RL setting with linear function approximation.
- **Simple reduction.**
  - At the initial state, if the agent chooses action a_1, then the agent receives a fixed reward value (say 0.5) and terminates.
  - If the agent chooses action a_2, then the agent transits to our hard instance. Therefore, in order to find a policy with suboptimality at most 0.5, the agent must evaluate the value of the optimal policy in our hard instance up to an error of 0.5, and hence the hardness result holds.

Upper bound: Low Distribution Shift or Policy Completeness are Sufficient

- **Notation**. For each $h \in [H]$, define

$$\Lambda_h = \mathbb{E}_{(s,a)\sim\mu_h}\left[\phi(s,a)\phi(s,a)^\top\right]$$

  to be the **feature covariance matrix of the data distribution at level** h.

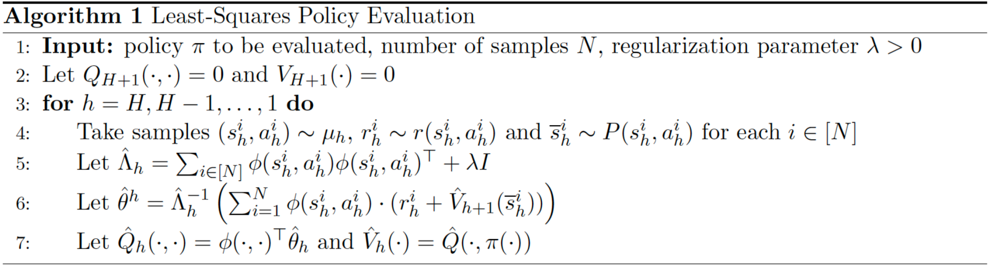
- Moreover, for each $h \in [H-1]$, define

$$\bar{\Lambda}_{h+1} = \mathbb{E}_{(s,a)\sim\mu_h,\bar{s}\sim P(\cdot|s,a)}\left[\phi(\bar{s},\pi(\bar{s}))\phi(\bar{s},\pi(\bar{s}))^\top\right]$$

  to be the **feature covariance matrix of the one-step lookahead distribution** induced by the data distribution at level h and π.

- Moreover, define $\bar{\Lambda}_1 = \phi\left(s_1, \pi\left(s_1\right)\right) \phi\left(s_1, \pi\left(s_1\right)\right)^\top$.
- We define Φ_h to be a $N \times d$ matrix, whose i-th row is $\phi\left(s_h^i, a_h^i\right)$, and define $\bar{\Phi}_{h+1}$ to be another $N \times d$ matrix whose i-th row is $\phi\left(\bar{s}_h^i, \pi\left(\bar{s}_h^i\right)\right)$.
- For each $h \in [H]$ and $i \in [N]$, define $\xi_h^i = r_h^i + V\left(\bar{s}_h^i\right) - Q\left(s_h^i, a_h^i\right)$.
- Clearly, $\mathbb{E}\left[\xi_h^i\right] = 0$ and $\left|\xi_h^i\right| \leq 2H$.
- We also use ξ_h to denote a vector whose i-th entry is ξ_h^i

LSPE Algorithm



Algorithm 1 Least-Squares Policy Evaluation

```
**Input:** policy $\pi$ to be evaluated, number of samples $N$, regularization parameter $\lambda > 0$
Let $Q_{H+1}(\cdot,\cdot) = 0$ and $V_{H+1}(\cdot) = 0$
**for** $h = H, H-1, \dots, 1$ **do**
    Take samples $(s_h^i, a_h^i) \sim \mu_h$, $r_h^i \sim r(s_h^i, a_h^i)$ and $\overline{s}_h^i \sim P(s_h^i, a_h^i)$ for each $i \in [N]$
    Let $\hat{\Lambda}_h = \sum_{i \in [N]} \phi(s_h^i, a_h^i)\phi(s_h^i, a_h^i)^\top + \lambda I$
    Let $\hat{\theta}^h = \hat{\Lambda}_h^{-1} \left( \sum_{i=1}^N \phi(s_h^i, a_h^i) \cdot (r_h^i + \hat{V}_{h+1}(\overline{s}_h^i)) \right)$
    Let $\hat{Q}_h(\cdot,\cdot) = \phi(\cdot,\cdot)^\top \hat{\theta}_h$ and $\hat{V}_h(\cdot) = \hat{Q}(\cdot, \pi(\cdot))$
```

The result of ordinary LSPE

Lemma 3.

Suppose $\lambda > 0$ in Algorithm 1, and for the given policy π, there exists $\theta_1, \theta_2, \ldots, \theta_d \in \mathbb{R}^d$ such that for each $h \in [H], Q_h^{\pi}(s,a) = \phi(s,a)^\top \theta_h$ for all $(s,a) \in \mathcal{S}_h \times \mathcal{A}$. Then we have

$$\left(Q^{\pi}\left(s_1, \pi\left(s_1\right)\right) - \hat{Q}\left(s_1, \pi\left(s_1\right)\right)\right)^2 = \left\|\sum_{h=1}^{H} \hat{\Lambda}_1^{-1} \Phi_1^\top \bar{\Phi}_2 \hat{\Lambda}_2^{-1} \Phi_2^\top \bar{\Phi}_3 \cdots \left(\hat{\Lambda}_h^{-1} \Phi_h^\top \xi_h - \lambda \hat{\Lambda}_h^{-1} \theta_h\right)\right\|_{\bar{\Lambda}_1}^2$$

Proof.

See Appendix B.1. □

Low distribution shift assumption

- **Low Distribution Shift.** The first special we focus on is the case where the distribution shift between the data distributions and the distribution induced by the policy to be evaluated is low.
- To measure the distribution shift formally, our main assumption is as follows.
- **Assumption** 3**.** We assume that for each $h \in [H]$, there exists $C_h \geq 1$ such that $\overline{\Lambda_h} \preceq C_h \Lambda_h$.

Upper bound under low distribution shift

Theorem 4.

Suppose for the given policy π, *there exists* $\theta_1, \theta_2, \ldots, \theta_d \in \mathbb{R}^d$ *such that for each* $h \in [H], Q_h^\pi(s,a) = \phi(s,a)^\top \theta_h$ *for all* $(s,a) \in \mathcal{S}_h \times \mathcal{A}$ *and* $\|\theta_h\|_2 \le H\sqrt{d}]^3$ *Let* $\lambda = CH\sqrt{d\log(dH/\delta)N}$ *for some* $C > 0$. *With probability at least* $1-\delta$, *for some* $c > 0$,

$$\left(Q_1^\pi\left(s_1, \pi\left(s_1\right)\right) - \hat{Q}_1\left(s_1, \pi\left(s_1\right)\right)\right)^2 \le c \cdot \left(\prod_{h=1}^{H} C_h\right) \cdot dH^5 \cdot \sqrt{\frac{d\log(dH/\delta)}{N}}$$

Proof.

See Appendix B.2. □

Illustration of error amplification on hard instance

- The factor $\prod_{h=1}^{H} C_h$ in implies that the estimation error will be amplified geometrically as the algorithm proceeds.
- If we run Algorithm 1 on the hard instance in Section 4, when $h = H$, the estimation error on $V\left(s_H^c\right)$ would be roughly $N^{-1/2}$ for each $c \in [\hat{d}]$.
- When using the linear predictor at level H to predict the value of s_H^*, the error will be amplified by $\hat{d}^{1/2}$.
- When $h = H - 1$, the dataset contains only s_{H-1}^c for $c \in [\hat{d}]$, and the estimation error on the value of s_{H-1}^c will be the same as that of s_H^*, which is roughly $(\hat{d}/N)^{1/2}$
- Again, the estimation error on the value of s_{H-1}^* will be $\left(\hat{d}^2/N\right)^{1/2}$ when using the linear predictor at level $H - 1$.
- As the algorithm proceeds, the error will eventually be amplified by a factor of $\hat{d}^{H/2}$, which corresponds to the factor $\prod_{h=1}^{H} C_h$ in Theorem 5.2

Explanation of $\sqrt{\hat{d}}$ amplification

- $\left(\hat{V}_H(s_H) - V_H(s_H)\right)^2$?

$$\hat{\theta}_H - \theta_H = \hat{\Lambda}_H^{-1}\left(\sum_{i=1}^{N}\phi(s_H^i, a_H^i)\left(r_H^i - r_H\right)\right)$$

$$\|\hat{\theta}_H - \theta_H\| \le \varepsilon_H \left\|\hat{\Lambda}_H^{-1}\left(\sum_{i=1}^{N}\phi(s_H^i, a_H^i)\right)\right\| \approx \varepsilon_H \left\|\frac{d}{N}I \cdot \frac{N}{d}\mathbf{1}\right\| = \varepsilon_H\sqrt{d}$$

- $\Lambda = \frac{1}{d}I$. Then $C \ge \frac{d}{2} = \hat{d}$ satisfies the condition $C\Lambda \succeq \overline{\Lambda}$, since

$$\overline{\Lambda} = \frac{1}{d}\left(\sum_{c\in[\hat{d}]} e_c\right)\left(\sum_{c\in[\hat{d}]} e_c\right)^T \succeq \frac{1}{d}\begin{bmatrix}\mathbf{1} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix} = \frac{1}{d}Q\,\mathrm{diag}\left(\frac{d}{2}, 0, \cdots, 0\right)Q^T$$

Policy Completeness Assumption

- **Assumption** 4. For the given policy π, for any $h > 1$ and $\theta_h \in \mathbb{R}^d$ with $\sup_{(s,a)\in\mathcal{S}_h\times\mathcal{A}} \left|\phi(s,a)^\top \theta_h\right| \leq H$, there exists $\theta' \in \mathbb{R}^d$ with $\|\theta'\|_2 \leq H\sqrt{d}$, such that for any $(s,a) \in \mathcal{S}_{h-1} \times \mathcal{A}$

$$\mathbb{E}[R(s,a)] + \sum_{s'\in\mathcal{S}_h} P\left(s' \mid s,a\right)\phi\left(s',\pi\left(s'\right)\right)^\top \theta_h = \phi(s,a)^\top \theta'$$

Results under policy completeness

- Under Assumption 4 and the additional assumption that the feature covariance matrix of the data distributions have lower bounded eigenvalue, i.e., $\sigma_{\min}(\Lambda_h) \geq \lambda_0$ for all $h \in [H]$ for some $\lambda_0 > 0$,
  - prior work [Chen and Jiang, 2019] has shown that for Algorithm 1, by taking $N = \mathrm{poly}(H, d, 1/\varepsilon, 1/\lambda_0)$ samples, we have $\left(Q_1^\pi(s_1, \pi(s_1)) - \hat{Q}_1(s_1, \pi(s_1))\right)^2 \leq \varepsilon$.
- The above analysis again implies that geometric error amplification is a real issue in offline RL, and sample-efficient offline RL is impossible unless
  - the distribution shift is sufficiently low, i.e., $\prod_{h=1}^{H} C_h$ is bounded,
  - or stronger representation condition such as policy completeness is assumed as in prior works [Szepesvári and Munos, 2005 , Chen and Jiang, 2019].

Outline

Introduction

Motivation

Offline RL Problems

Statistical limits

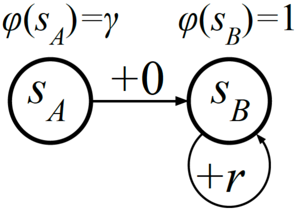
Finite horizon episodic setting

Infinite horizon discounted setting

Notations for infinite horizon discounted setting

- Wang, Foster, and Kakade [2020] recently showed that in finite-horizon batch RL, the sample complexity of evaluating a given policy π has an information-theoretic lower bound that is exponential in the horizon, even if realizable linear features are given
  - i.e., $\varphi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ such that $Q^\pi(\cdot) = \langle \varphi(\cdot), \theta^\pi \rangle$ for some parameter $\theta^\pi \in \mathbb{R}^d$)
  - and data provides good feature coverage (i.e., $\mathbb{E}\left[\varphi\varphi^\top\right]$ has lower-bounded eigenvalues under the data distribution).
- Amortila, Jiang, and Xie [2020] show that its analogy in the discounted setting has a stronger statement (infinite sample complexity) with a simpler construction (1-d feature, 2 states, and arbitrary discount factor.)

Construction



- Consider the deterministic MDP in Figure with a discount factor $\gamma \in (0,1)$, where every state only has 1 action (which we omit in the notations).
- s_A transitions to s_B with 0 reward, and s_B has a self-loop with r reward per step.
- The batch data only contains the tuple $(s_A, 0, s_B)$.
- The feature map is 1-dimensional: $\varphi(s_A) = \gamma$ and $\varphi(s_B) = 1$.
- Clearly, without data from s_B, the learner cannot know the value of r, hence cannot determine the value of s_A or s_B, even with an infinite amount of data.

Verifying realizabilty assumptions

- **Realizability**: We show that $V^{\pi}(\cdot) = \langle \varphi(\cdot), \theta \rangle$ for some $\theta \in \mathbb{R}$.
- By the Bellman equation, $V^{\pi}\left(s_A\right) = \gamma V^{\pi}\left(s_B\right)$.
- Therefore, $V^{\pi}\left(s_A\right) = \langle \varphi\left(s_A\right), V^{\pi}\left(s_B\right) \rangle$.
- Similarly $V^{\pi}\left(s_B\right) = 1 \cdot V^{\pi}\left(s_B\right) = \langle \varphi\left(s_B\right), V^{\pi}\left(s_B\right) \rangle$.
- So $V^{\pi}(\cdot)$ is always linearly-realizable, with $\theta = V^{\pi}\left(s_B\right) = \frac{r}{1-\gamma}$ being the unknown coefficient.

Verifying coverage assumptions

- **Coverage**: Translating the condition of Wang et al. (2020) to the discounted case, it is required that: (1) $\|\varphi(\cdot)\|_2 \leq 1$ always holds, and (2) $\mathbb{E}\left[\varphi\varphi^\top\right]$ has polynomially lower bounded eigenvalues.
- (1) is satisfied in our construction.
- (2) since we only have data from s_A, the feature covariance matrix under the data distribution is $\varphi\left(s_A\right)\varphi\left(s_A\right)^\top = \gamma^2$, whose only eigenvalue is γ^2 and is well above 0 as long as γ is.

Extensions for general d

- Although it is sufficient to prove the lower bound for $d = 1$, the construction easily scales to arbitrary d :
- We simply make d copies of the construction in Figure 1, and assign a coordinate of $\varphi : \mathcal{S} \to \mathbb{R}^d$ to each copy. Let data be uniform over the s_A of all copies, so the feature covariance matrix is $\gamma^2/d \cdot I$

$$\phi(s_A^i) := \begin{bmatrix} 0 \\ \vdots \\ \gamma \\ 0 \\ \vdots \end{bmatrix} \qquad (i - \text{th component})$$

Extensions for general d and the controlled setting

- The extension to the controlled case is similar.
- Let a denote the action of s_A in Figure.
- We introduce a second action a' for s_A that transitions to s_C with 0 reward, and s_C is absorbing with reward r'.
- Let the 2-dimensional feature map be: $\varphi\left(s_A, a\right) = [\gamma, 0]^\top$, $\varphi\left(s_A, a'\right) = [0, \gamma]^\top, \varphi\left(s_B\right) = [1, 0]^\top, \varphi\left(s_C\right) = [0, 1]^\top$.
- It is easy to verify that $Q^\star$ is realizable by take $\theta = [\frac{r}{1-\gamma}, \frac{r'}{1-\gamma}]$.
- However, $Q^\star\left(s_A, a\right) = \frac{\gamma}{1-\gamma} r$ and $Q^\star\left(s_A, a'\right) = \frac{\gamma}{1-\gamma} r'$ can independently take arbitrary values between $[0, \gamma/(1-\gamma)]$ (assuming rewards lie in [0,1]), so the learner cannot choose a near-optimal action even with infinite data.

Lower bound proposition

- **Proposition 1 (Informal).** For any $d \geq 1, \gamma \in (0, 1)$, given realizable linear features, the value function learned by any batch RL algorithm must have $\Omega(1)$ worst-case error, even with an infinitely large dataset that has $\Theta(1/d)$ feature coverage.

Final Remark

- While the discounted setting allows a very simple construction for the lower bound, this does not imply that the construction for the finite-horizon setting can be simplified in a similar manner.
- In fact, we believe that the careful construction of [Wang et al., 2020] that cleverly exponentiates a negligibly small error is necessary for the finite horizon setting.
- Such a difference between the finite-horizon setting and the discounted setting, however, does challenge the conventional wisdom that the results in the finite horizon setting and the discounted setting are often similar and translate to each other with $H = O(1/(1-\gamma))$ up to minor differences.
- Are these two lower bounds "essentially the same", or does their difference imply some fundamental difference between the finite-horizon and the discounted settings?

References I

P. Amortila, N. Jiang, and T. Xie. A variant of the wang-foster-kakade lower bound for the discounted setting. arXiv preprint arXiv:2011.01075, 2020.

J. Chen and N. Jiang. Information-theoretic considerations in batch reinforcement learning. arXiv preprint arXiv:1905.00360, 2019.

D. Hsu, S. M. Kakade, and T. Zhang. Random design analysis of ridge regression. Foundations of Computational Mathematics, 14(3):569–600, 2014.

A. Kumar, J. Fu, M. Soh, G. Tucker, and S. Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In Advances in Neural Information Processing Systems, pages 11761–11771, 2019.

S. Levine, A. Kumar, G. Tucker, and J. Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv preprint arXiv:2005.01643, 2020.

References II

Y. Luo, H. Xu, and T. Ma. Learning self-correctable policies and value functions from demonstrations with negative sampling. arXiv preprint arXiv:1907.05634, 2019.

R. Wang, D. P. Foster, and S. M. Kakade. What are the statistical limits of offline rl with linear function approximation? arXiv preprint arXiv:2010.11895, 2020.

Y. Wu, G. Tucker, and O. Nachum. Behavior regularized offline reinforcement learning, 2019.