Towards Practical Offline Reinforcement Learning: Sample Efficient Policy Selection and Evaluation

by

Vincent Liu

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Statistical Machine Learning

Department of Computing Science

University of Alberta

Abstract

Offline reinforcement learning (RL) involves learning policies from datasets, rather than online interaction. The dissertation first investigates a critical component in offline RL: offline policy selection (OPS). Given that most offline RL algorithms require careful hyperparameter tuning, we need to select the best policy amongst a set of candidate policies before deployment. In the first part of the dissertation, we provide clarity on when OPS is sample efficient by building a clear connection to off-policy policy evaluation (OPE) and Bellman error estimation. This dissertation then presents algorithms to leverage offline data. We begin by examining environments that include exogenous variables with limited agent impact and endogenous variables under full agent control. We show that policy evaluation and selection become straightforward under such conditions. Additionally, we present an algorithm based on Fitted-Q Iteration with data augmentation and show its ability to find nearly optimal policies with polynomial sample complexity. We then study OPE in non-stationary environments and introduce the regression-assisted doubly robust estimator, which effectively incorporates the past data without introducing a large bias and improves on existing OPE estimators with the use of auxiliary information and a regression approach. We evaluate our algorithms across a variety of problems, some built using real-world datasets, including optimal order execution, inventory management, hybrid car control and recommendation systems.

Preface

This dissertation contains three original works by the author, including two published works "Asymptotically Unbiased Off-Policy Policy Evaluation when Reusing Old Data in Nonstationary Environments" (V. Liu, Chandak, et al., 2023) and "Exploiting Action Impact Regularity and Exogenous State Variables for Offline Reinforcement Learning" (V. Liu, Wright, et al., 2023), and one under submission "When is Offline Policy Selection Sample Efficient for Reinforcement Learning?".

Acknowledgements

I am deeply grateful to my amazing advisor, Martha White, for her support, encouragement, and invaluable guidance. Her expertise and knowledge have been instrumental in shaping me as a researcher and her kindness always reminds me to be kind to everyone.

I am also grateful for the valuable feedback provided by my supervisory committee members and PhD examination members (in alphabetical order): Emma Brunskill, Csaba Szepesvári, Xiaoqi Tan, Matthew Taylor, James Wright. I extend my sincere thanks to Philip Thomas, Adam White, and James Wright for advising on my research projects, and Rohana Karunamuni on teaching several statistics courses that are helpful for my research. I also feel very fortunate to learn from Csaba about the theoretical foundations of bandit, reinforcement learning, and machine learning. I would also like to thank my undergraduate advisor, Shou-De Lin, who introduced me to machine learning research.

I want to thank all my collaborators: Alex Ayoub, Alan Chan, Yash Chandak, Khurram Javed, Prabhat Nagarajan, Somjit Nath, Andy Patterson, David Tao, Han Wang, and internship mentors: Paul Aubin, Daniel Jiang, Bo Meng and Hengshuai Yao. I am truly grateful for the knowledge and experience I gained from working with them. I am particularly grateful to Yash for his constructive feedback and advice. I would like to thank my wonderful RLAI, Amii and UofA friends, who have made this journey truly enjoyable.

Thank my family for their constant support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Reinforcement learning (RL) is a framework for learning to make sequential decisions by interacting with an unknown environment. At each time step, the agent chooses an action based on its decision-making rule, which is called a policy, and receives an immediate reward and the next observation from the environment. The goal of the agent is learn a policy that maximizes the cumulative rewards while interacting the environment.

There has been significant progress for RL on a number of simulated environments, such as video games (Mnih et al., 2015; Vinyals et al., 2019) and and board games (Silver et al., 2017). However, the need for a large amount of interactions prohibits many other real-world applications, beyond simulated environments. This limitation arises for several reasons:

- Infeasibility of online interactions: In some cases, online interactions with real-world environments is not feasible, as it can be unsafe or even dangerous. For instance, when it comes to self-driving cars, direct interactions with the real-world environment may carry significant risks.

- Costly simulator development: Building a simulator that accurately represents a real-world environment can be costly.

- Expense of simulation: Even when a simulator is available, the process of simulation itself can be expensive in terms of computation.

For many real-world problems, we need to consider using previously collected dataset, without further online interaction or simulation, to find a good

policy. This problem setting is called the *offline* setting. By leveraging static datasets, offline RL can bridge the gap between the remarkable success of RL in simulated environments and its practical applicability in the real world.

Offline RL problem is known to be hard in general (Chen & Jiang, 2019). That is, without assumptions on the data distribution and the environment dynamics, one can easily construct an adversarial example such that the agent cannot learn a good policy from a limited dataset with high probability. Several works have made progress toward understanding *when* offline RL is feasible. For example, it has been shown that a good data distribution alone is not sufficient for sample efficient offline learning (Chen & Jiang, 2019; R. Wang et al., 2021), so we can not expect offline learning to work in all situations even when we can choose the data distribution. We must consider assumptions on the environments. On the other hand, a good data coverage, which is often measured by the concentration coefficient (Munos, 2007), together with a function class that has a low inherent Bellman error (Antos et al., 2008) enables sample efficient learning (Chen & Jiang, 2019).

These theoretical results provide useful insights on the conditions under which offline RL works; however, there are still many unsolved challenges in practice. First, these theoretical results often relies on assumptions such as a small concentration coefficient, which are unrealistic, as we will discuss more in Chapter 3. Moreover, even if we can derive a bound on the performance loss of the output policy, the bounds often depend on quantities such as the concentration coefficient or the inherent Bellman error that are hard to compute exactly. In practice, we still need to verify whether the output policy can actually perform well in the underlying environment, and we often also need to tune the hyperparameters to find a good output policy. These practical challenges are still not fully addressed by existing theoretical or empirical works.

To address these challenges in the offline setting, we focus on off-policy policy evaluation (OPE) and offline policy selection (OPS) in the this dissertation. OPE is the problem where we aim to estimate the expected return of a given target policy. It allows us to evaluate the performance of the output

policy with the data collected from a different policy. If we can perform OPE accurately, we can use the estimators to check whether the policy is safe before deployment (P. Thomas et al., 2015b). Similarly, OPS is the problem where we aim to select the best-performing policy from a set of candidate policies. With the help of a good OPS method, we can perform hyperparameter tuning, unlocking the full potential of numerous state-of-art offline policy learning algorithm.

The goal of the thesis is to design practical offline RL algorithms under real-world scenarios. In most real-world applications, we often have limited data, so achieving sample efficiency becomes an important factor in determining the practicality of offline RL algorithms. Moreover, considering the deployment of offline RL algorithms, it is critical to ensure the safety and reliability of the policy before deploying it. As a result, we place a strong emphasis on providing sample efficiency and performance guarantees.

1.1 Contributions

I list the main contributions included in the dissertation.

1.1.1 Towards sample efficient offline policy selection

In the first part of the dissertation, we study the offline policy selection problem. Policy selection is a critical procedure before deploying RL algorithms in real-world applications. However, there is little understanding about the fundamental limitations of the offline policy selection (OPS) problem. In this part, we aim to provide clarity on when sample efficient OPS is possible, primarily by bringing together existing theoretical results about off-policy policy evaluation (OPE) and Bellman error (BE) estimation. In particular, we first show a negative result, that in the worst case, OPS is just as hard as OPE, by proving a reduction of OPE to OPS. As a results, no OPS method can be more sample efficient than OPE in the worse case. Then we propose a simple BE estimation method for OPS. We highlight that using BE generally has more requirements, but if satisfied, has an easy method for selecting its own hyper-

parameters and might be more sample efficient than OPE. We conclude with an empirical study comparing OPE and BE estimation for OPS in benchmark environments.

1.1.2 Environments with exogenous structure

In the second part of the dissertation, we explore a restricted class of MDPs to obtain guarantees for offline policy learning and policy evaluation. The key property, which we call Action Impact Regularity (AIR), is that actions primarily impact the endogenous state and have limited impact on the remaining part of the state (the exogenous component). We also assume we have the knowledge of the endogenous dynamics. The assumption is strong, however, it holds in a number of real-world domains including financial and operation research problems. We propose and theoretically analyze an algorithm based on Fitted-Q Iteration that exploits the property. We empirically show the proposed algorithm outperforms existing offline RL algorithms in several synthetic and real-world environments that satisfy the AIR property.

1.1.3 Environments with nonstationarity

In the third part of the dissertation, we study the OPE problem for contextual bandits and finite horizon reinforcement learning in non-stationary environments, where reusing old data is critical for sample efficient policy evaluation. We introduce a variant of the doubly robust (DR) estimator, called the regression-assisted DR estimator, that reuses the old data without introducing a large bias and unifies several existing off-policy evaluation methods. We prove that the estimator has asymptotic properties, in particular, we can construct a large sample confidence interval for the true policy value. We empirically show the proposed estimator provide tight and valid confidence intervals in several recommendation environments.

1.2 Overview

The remainder of the dissertation is organized as follows. In Chapter 2, we introduce the necessary background knowledge that will be used throughout the entire thesis. In Chapter 3, we provide a summary of existing theoretical results. Chapter 4 studies the sample complexity of OPS. Chapter 5 presents the FQI-AIR algorithm for environments with exogenous structure. Chapter 6 presents the regression-assisted estimator for nonstatinoary environment. We conclude in Chapter 7, with a discussion on the limitation of the thesis and future directions.

Chapter 2

Background

We provide the background knowledge in this section. The purpose of this section is to provide a brief introduction of reinforcement learning and offline reinforcement learning that will be used throughout the entire dissertation, and detailed problem settings and notation will be introduced in the following chapters.

2.1 Value functions and Bellman operators

We formalize the agent-environment interaction as a finite horizon Markov decision process (MDP) $M = (\mathcal{S}, \mathcal{A}, Q, H, \nu)$ where $\mathcal{S}$ is a set of states, and $\mathcal{A}$ is an set of actions; for simplicity, we assume that both sets are finite but they can be very large. $H \in \mathbb{Z}^+$ is the planning horizon, and $\nu \in \Delta(\mathcal{S})$ the initial state distribution. Given a state s and an action a, the reward R and next state S' are sampled from a stochastic kernel Q, that is, $(R, S') \sim Q(\cdot|s, a)$. We assume the reward is bounded in $[0, r_{max}]$ almost surely, so the total return of each episode is bounded in $[0, V_{max}]$ with $V_{max} = Hr_{max}$. The stochastic kernel Q induces a transition probability $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, and a mean reward function $r : \mathcal{S} \times \mathcal{A} \to [0, r_{max}]$ which gives the mean reward when taking action a in state s.

In the finite horizon setting, the policies are non-stationary. A non-stationary policy is a sequence of memoryless policies $(\pi_0, \ldots, \pi_{H-1})$ where $\pi_h : \mathcal{S} \to \Delta(\mathcal{A})$. We assume that the set of states reachable at time step h, $\mathcal{S}_h \subset \mathcal{S}$, are disjoint, without loss of generality, because we can always define a new state

space $\mathcal{S}' = \mathcal{S} \times [H]$ where $[n] := \{0, 1, 2, \ldots, n-1\}$. Then, it is sufficient to consider stationary policies $\pi : \mathcal{S} \to \Delta(\mathcal{A})$.

Value functions. Given a policy π, $h \in [H]$, and $(s, a) \in \mathcal{S} \times \mathcal{A}$, we define the value function and the action-value function as

$$v_h^\pi(s) := \mathbb{E}^\pi \left[\sum_{t=h}^{H-1} r(S_t, A_t) | S_h = s \right]$$

and

$$q_h^\pi(s, a) := \mathbb{E}^\pi \left[\sum_{t=h}^{H-1} r(S_t, A_t) | S_h = s, A_h = a \right]$$

where the expectation is with respect to $\mathbb{P}_M^\pi$ (we may drop the subscript M when it is clear from the context). $\mathbb{P}^\pi$ is the probability measure on the random element $(S_0, A_0, \ldots, S_{H-1}, A_{H-1})$ induced by the policy π and the MDP such that $\mathbb{P}^\pi(S_0 = s) = \nu(s)$, $\mathbb{P}^\pi(A_t = a | S_0, A_0, \ldots, S_t) = \pi(a|S_t)$, and $\mathbb{P}^\pi(S_{t+1} = s' | S_0, A_0, \ldots, S_t, A_t) = P(S_t, A_t, s')$ for $t \geq 0$ (Lattimore & Szepesvári, 2020; Puterman, 2014).

Given a policy π, we define the value of policy as $J(\pi) = \mathbb{E}_{s_0 \sim \nu}[v_0^\pi(s_0)]$. The visitation distribution (or sometimes called occupancy measure) of π at time step h is given by $d_h^\pi(s, a) = \mathbb{P}^\pi(S_h = s, A_h = a)$, and $d^\pi(s, a) = \frac{1}{H} \sum_h \mathbb{P}^\pi(S_h = s, A_h = a)$.

A policy π is greedy with respect to the value function $q = (q_0, \ldots, q_{H-1})$ if $\pi(a|s_h) = \arg\max_{a \in \mathcal{A}} q_h(s_h, a)$ for any $h \in [H]$ and $s_h \in \mathcal{S}_h$. The optimal action-value function is defined by $q_h^*(s) := \sup_\pi q_h^\pi(s)$. A policy achieving this is called an optimal policy. The fundamental theorem of MDPs states that the greedy policy with respect to $q^* = (q_0^*, \ldots, q_{H-1}^*)$ is optimal, so knowing the optimal action-value function is sufficient to behave optimally in the MDP.

Similar to the stationary policy, we also use an action-value function q to denote the sequence of action-value function $(q_0, \ldots, q_{H-1})$ since the set of states reachable at each time step are disjoint. That is, for all $h \in [H]$ and $s \in \mathcal{S}_h$, $q(s, \cdot) = q_h(s, \cdot)$.

Bellman operators. The Bellman operator is defined by

$$(\mathcal{T}q_h)(s,a) := r(s,a) + \sum_{s' \in \mathcal{S}} P(s,a,s') \max_{a' \in \mathcal{A}} q_h(s',a').$$

The Bellman evaluation operator for a policy π is defined by

$$(\mathcal{T}^\pi q_h)(s,a) := r(s,a) + \sum_{s' \in \mathcal{S}} P(s,a,s') \sum_{a' \in \mathcal{A}} \pi(a'|s') q_h(s',a').$$

Most reinforcement learning algorithms can be viewed as applying the Bellman operator iteratively to find the optimal action-value function. *Value Iteration* (VI) computes a sequence of action-value function by applying the Bellman operator. Let $q_H = 0$ and, for $h = H-1, \ldots, 0$, let

$$q_h = \mathcal{T} q_{h+1}.$$

It can be verified that the output action-value function $q = (q_0, \ldots, q_{H-1})$ is optimal.

Policy Iteration (PI) alternates between the policy evaluation phase and the policy improvement phase. Starting with an arbitrary policy π, PI first computes a sequence of action-value function by applying the Bellman evaluation operator. Let $q_H = 0$, and for $h = H-1, \ldots, 0$, let

$$q_h = \mathcal{T}^\pi q_{h+1}.$$

It can also be verified that that the output action-value function $q = (q_0, \ldots, q_{H-1})$ is the true action-value function q^π. Once the true action value function is found, the policy improvement theorem shows that we can obtain a better policy by greedifying q^π. The procedure is repeated until the optimal solution (q^*, π^*) is found.

2.2 Value function approximation

In the previous part, we introduced Value Iteration and Policy Iteration. To perform these methods exactly, we need to store individual values for each state-action pair. This requirement for storing individual values can result in a substantial memory cost, especially when dealing with a large state space.

In practice, we often approximate the action-value function. For example, we can use a model $f_\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ to represent the action-value function where $\theta \in \Theta$ is the parameter of the model, and the set of functions over the state-action space is

$$\mathcal{F} = \{f_\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R} \mid \theta \in \Theta\}.$$

To ensure the approximated function is still accurate, we need some assumptions on the function class. If our goal is to find the optimal action-value function, we need a function class such that $q_0^*, \ldots, q_{H-1}^* \in \mathcal{F}$. We call a function class q^*-realizable if it satisfies the condition. Similarly, we say a function class is q^π-realizable if $q_0^\pi, \ldots, q_{H-1}^\pi \in \mathcal{F}$.

Consider Value Iteration with function approximation, since we might not be able to represent the function $\mathcal{T} q_h$ in the function class, we need to project the function back to the function class $\mathcal{F}$. This can be done by a projection operator. Let $q \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$, we define

$$\Pi_{\mathcal{F}} q := \arg\min_{f \in \mathcal{F}} \|f - q\|_{p,\nu}^p,$$

where $\|q\|_{p,\nu}^p = \sum_{s \in \mathcal{S}, a \in \mathcal{A}} \nu(s,a) |q(s,a)|^p$ is a weighted norm of q. There are many choices of the distribution ν. Most commonly, if we have access to a set of representable state-action pairs D, then we can consider the weighting under the empirical distribution that puts a point mass to each state-action pair in D, that is, $\|q\|_{p,D}^p = \frac{1}{|D|} \sum_{(s,a) \in D} |q(s,a)|^p$

The resulting algorithm is called *Approximate Value Iteration*, since we approximate the Bellman operator. Let $q_h = 0$ and

$$q_h = \Pi_{\mathcal{F}} \mathcal{T} q_{h+1}.$$

We are not guaranteed to find the optimal solution with function approximation. However, there exists results that relate the supoptimiality of the approximated solution to the approximation power of the function class (Munos, 2005).

Algorithm 1 Fitted Q Evaluation

1: Input: dataset D and function class $\mathcal{F}$
2: $q_H = 0$
3: **for** $h = H-1, \ldots, 0$ **do**
4: $\quad q_h = \arg\min_{q \in \mathcal{F}} \sum_{(s,a,r,s') \in D_h} (q(s,a) - r - \sum_{a' \in \mathcal{A}} \pi(a'|s') q_{h+1}(s', a'))^2$
5: Output: $\hat{J}(\pi) = \frac{1}{n} \sum_{s_0 \in D} q_0(s_0)$

2.3 Offline reinforcement learning

In the offline setting, we are given a fixed set of transitions

$$D = \{(S_i, A_i, R_i, S_i')\}_{i=1}^{nH}$$

with (S_i, A_i) drawn from a data distribution μ over $(\mathcal{S} \times \mathcal{A})$, and (R, S') sampled from the stochastic kernel Q. In this thesis, we consider the data collection scheme where the data is collected by a data collection policy π_b. That is, D consists of n trajectories

$$\tau^{(i)} = (S_0^{(i)}, A_0^{(i)}, R_0^{(i)}, \ldots, S_{H-1}^{(i)}, A_{H-1}^{(i)}, R_{H-1}^{(i)})$$

induced by the the interaction of the policy π_b and the MDP M. We also use D_h to denote the data at horizon h. The data distribution $\mu_h(s,a) = d_h^{\pi_b}(s,a)$ is the visitation distribution of the data collection policy.

Policy evaluation. Given a target policy π, the goal of policy evaluation is to estimate the value of the policy π, which is defined as $J(\pi) := \mathbb{E}_{S_0 \sim \nu}[v_0^\pi(S_0)]$. A foundational strategy to estimate $J(\pi)$ is to use the importance sampling (IS) estimator, which corrects the sampled returns seen in the dataset by the product of IS ratios:

$$\hat{J}_{\mathrm{IS}}(\pi) = \frac{1}{n} \sum_{i=1}^{n} \left(\prod_{h=0}^{H-1} \frac{\pi(A_h^{(i)}|S_h^{(i)})}{\pi_b(A_h^{(i)}|S_h^{(i)})} \right) \left(\sum_{h=0}^{H-1} R_h^{(i)} \right).$$

The IS estimator is an unbiased estimator for $J(\pi)$, as long as $\pi_b(a|s) > 0$ for any (s,a) such that $\pi(a|s) > 0$. However, the estimator can have high variance.

Another strategy is to applies the projected Bellman evaluation operator to learn the action-value function. In the offline setting, this method is called

Algorithm 2 Fitted Q Iteration

Input: dataset D and function class $\mathcal{F}$
$q_H = 0$
for $h = H-1, \ldots, 0$ **do**
  $q_h = \arg\min_{q \in \mathcal{F}} \sum_{(s,a,r,s') \in D_h} (q(s,a) - r - \max_{a' \in \mathcal{A}} q_{h+1}(s', a'))^2$
  $\pi_h(s) = \arg\max_{a \in \mathcal{A}} q_h(s,a)$ for all $s \in \mathcal{S}_h$
Output: $\pi = (\pi_0, \ldots, \pi_{H-1})$

Fitted Q evaluation (FQE) (Le et al., 2019), described in Algorithm 1. The FQE estimator has low variance but potentially high bias if we don't choose the function class carefully.

Note that in the offline setting, we do not know the transition probability and the mean reward function. Therefore, we cannot compute the projected Bellman evaluation operator exactly. We need to approximate it with samples, that is,

$$\Pi_{\mathcal{F}} \mathcal{T}^{\pi} q \approx \arg\min_{f \in \mathcal{F}} \frac{1}{|D|} \sum_{(s,a,r,s') \in D} |f(s,a) - r - \sum_{a' \in \mathcal{A}} \pi(a'|s') q(s', a')|^2.$$

Policy learning. For policy learning, the goal is to learn an optimal policy from the offline data. A representative algorithm for offline policy learning is *Fitted Q Iteration* (FQI) (Ernst et al., 2005), described in Algorithm 2. Similar to FQE, FQI applies the projected Bellman operator to learn a sequence of action-value function $q = (q_0, \ldots, q_{H-1})$, and output the greedy policy with respect to q.

In this thesis, our primary emphasis is on algorithms based on the use of the Bellmen operators, such as FQE and FQI. These algorithms fall under the category of approximate dynamic programming (ADP). Additionally, there exist alternative algorithms designed to learn different quantities. For example, model-based methods (Mannor et al., 2007) aim to learn a model of the environment, which includes estimating the transition probability and the reward function. Marginalized IS methods (Q. Liu et al., 2018; Uehara et al., 2020; Xie et al., 2019) along with DICE methods (Nachum et al., 2019) focus on estimating the density ratio between the visitation distribution of a target policy and the data collection policy. We have chosen to focus on ADP

methods since they already offer a comprehensive framework well-suited to address the challenges in offline RL that are central in this thesis.

Performance measure. To assess the quality of policy learning algorithms, we measure the suboptimality (which is also called the performance loss or the regret) of the policy. The suboptimality is the difference between the value of an optimal policy and the value of the output policy, that is, $J(\pi^*) - J(\pi)$. Similarly, for policy evaluation, we can measure the distance between the true value and the estimator: $|J(\pi) - \hat{J}(\pi)|^p$ for some $p \geq 1$. For $p = 1$, we consider the absolute error. For $p = 2$, we consider the squared error.

We are interested in finite sample guarantees in the PAC learning framework. Suppose we run FQI with a finite function class $\mathcal{F}$ and find an output policy $\hat{\pi}$. We want to guarantee that $J(\pi^*) - J(\hat{\pi}) \leq \varepsilon$ with probability at least $1 - \delta$ using a sample size less than some polynomial function of $1/\varepsilon$, $1/\delta$, $\log |\mathcal{F}|$, and other relevant quantities such as H. We might derive asymptotic guarantees instead of finite sample guarantees. For example, we can show that $J(\pi^*) - J(\hat{\pi}) \to 0$ in probability and the rate of convergence $J(\pi^*) - J(\hat{\pi}) = o_P(n^\alpha)$. Detailed performance measures will be introduced when needed.

Chapter 3

The Practicality of Offline RL

"There is nothing as practical as a good theory"
— Kurt Lewin

The goal of the chapter is to provide a summary on existing theoretical results and their insights for the practicality of offline RL. We begin the chapter by discussing the hardness of the offline setting, then we discuss conditions that enable sample efficient offline learning.

Achieving sample efficiency is an important factor in determining the practicality of offline RL algorithms. While we acknowledge the importance of computational efficiency as an additional practical consideration, our primary focus revolves around sample complexity. In finite horizon offline RL, sample complexity is a measure that informs us about the smallest number of episodes an algorithm requires to find a nearly optimal policy. We classify an algorithm as sample efficient if its sample complexity does not depend on the size of the state space $|\mathcal{S}|$ and does not grow exponentially with other relevant factors, such as the planning horizon H and number of actions $|\mathcal{A}|$.

3.1 When does offline RL not work?

The challenge in offline RL is that the quality of the output policy can be highly dependent on the data, the underlying environment, and the function approximation. Most obviously, the data might not cover some parts of the environment, resulting in two issues. The first is that the learned policy, when executed in the environment, is likely to deviate from the behavior that

generated its training data and reach a state-action pair that was unseen in the dataset. For these unseen state-action pairs, the algorithm has no information about how to choose a good action. The second issue is that if the dataset does not contain transitions in the high-reward regions of the state-action space, it may be impossible for any algorithm to return a good policy. One can easily construct a family of MDPs with missing data such that no algorithm can identify the MDP and suffer a large suboptimality gap (Chen & Jiang, 2019).

The data distribution used to collect the offline data (in our notation, μ) is a critical component, however, a good data distribution alone is not sufficient. We also need assumptions on the underlying MDPs. In particular, Chen and Jiang (2019) showed that if we do not make assumptions on the MDP dynamics, no algorithm can achieve a polynomial sample complexity to return a near-optimal policy, even when the algorithm can choose any data distribution. R. Wang et al. (2021) provide an exponential lower bound for the sample complexity of off-policy policy evaluation (for a given policy π) under linear function approximation, even with q^π-realizable function class and when the data distribution induces a well-conditioned covariance matrix. Zanette (2020) provide an example where offline RL is exponentially harder than online RL, even with the best data distribution, q^π-realizable function class and assuming the exact feedback is observed for each sample in the dataset. Xiao et al. (2022) provide an exponential lower bound for the sample complexity of obtaining nearly-optimal policies when the data is obtained by following a data collection policy. All these results suggest designing a good data distribution to collect data is not sufficient.

Additionally, offline RL requires a strong representational condition on function approximation. In supervised learning, the standard realizability condition asserts that the function class can represent the optimal function to be learned, such as the conditional mean in regression tasks. Similarly, in offline RL, we can consider an analogous optimal condition such that the function class contains the optimal value function.

Nonetheless, a study by Foster et al. (2021) shows that this realizability

condition, even when combined with standard data coverage requirement,[1] is insufficient. These findings suggest that, in the context of offline RL, additional conditions beyond realizability is needed.

One such condition is the Bellman completeness condition, which asserts that the function class should be capable of representing the Bellman operator for all the value function within the class. That is, for any $q \in \mathcal{F}$, it should hold that $\mathcal{T}q \in \mathcal{F}$. It's worth noting that the Bellman completeness condition lacks a monotonic property: when we increase the complexity or the size of the function class, it is possible that condition become unsatisfied. Hence, increasing the size of the function class is not always a viable solution for satisfying this condition.

An ongoing research direction is to explore the conditions under which realizability alone suffices for offline RL. Zanette (2022) investigate the sufficiency of realizability through a localized version of Bellman completeness. Furthermore, Xie and Jiang (2021) propose an algorithm that relies solely on realizability, but it comes with more stringent data coverage requirements and is computationally intractable.

These challenges in offline RL seem to resemble the well-known deadly triad issue (Sutton & Barto, 2018), though is different in an important way. The deadly triad issue states that convergence of an RL algorithm is not guaranteed when off-policy learning, bootstrapping, and function approximation are combined. Specifically, the data coverage condition relates to the off-policy aspect, while the representational condition pertains to bootstrapping and function approximation. In the context of offline RL, our primary concern is the ability to find nearly-optimal policies from offline data. The algorithm itself does not need to be convergent (Chen & Jiang, 2019).

The message derived from these findings is that offline RL presents inherent challenges. It is necessary to consider problem-specific assumptions about the data, the underlying MDPs, and function approximation. This forms the principal motivation behind the thesis.

[1] We will discuss the data coverage requirement in the following subsection.

3.2 When does offline RL work?

In this section, we provide a summary of positive results and discuss how we can obtain guarantees for offline RL under specific conditions.

3.2.1 Warm-up: offline contextual bandit

We begin by examining the feasibility of offline learning for contextual bandit (CB) problems. CB can be seen as a special case of RL with a short horizon $H = 1$. As the horizon is short, we are not concerned with the exponential lower bound mentioned in the previous section, making offline learning in the context of CB relatively straightforward.

For contextual bandit with logged data, we can employ importance sampling (IS) for policy evaluation. For $H = 1$, the IS estimator is

$$\hat{J}(\pi) = \frac{1}{n} \sum_{(s,a,r) \in D} \frac{\pi(a|s)}{\pi_b(a|s)} r.$$

We can then perform policy optimization using this estimate within a specified policy class Π:

$$\hat{\pi} = \arg\min_{\pi \in \Pi} \hat{J}(\pi).$$

This approach is often referred to as *counterfactual risk minimization* (Swaminathan & Joachims, 2015a). To prevent overfitting, we can incorporate a sample variance term using empirical Bernstein bounds (Maurer & Pontil, 2009).

We only require a mild condition in which the behavior policy covers the target policy. In other words, if $\pi(a|s) > 0$ for some $(s, a) \in \mathcal{S} \times \mathcal{A}$, then $\pi_b(a|s) > 0$. This condition can be met by introducing randomness in the data collection policy.

In applications involving a large number of actions, such as recommendation systems, fulfilling this assumption can be challenging. To address the lack of coverage, particularly in scenarios with large action spaces, Sachdeva et al. (2020) offer various approaches to enable offline learning. These approaches include action space restriction, policy space restriction, and the utilization of reward extrapolation.

3.2.2 Data coverage conditions

Now we explore the data coverage conditions required to obtain guarantees for offline RL, extending beyond CB. We can naturally apply IS, similar to what's done in CB, as long as the data collection policy is randomized. In this case, we aim to find the best policy $\hat{\pi}$ within the policy class Π:

$$\hat{\pi} = \arg\max_{\pi \in \Pi} \frac{1}{n} \sum_{i=1}^{n} \left(\prod_{h=0}^{H-1} \frac{\pi(A_h^{(i)}|S_h^{(i)})}{\pi_b(A_h^{(i)}|S_h^{(i)})} \right) \left(\sum_{h=0}^{H-1} R_h^{(i)} \right).$$

However, the IS estimator comes with exponentially high variance. This implies that, in the worst case, we may require an exponential number of episodes to learn an effective policy.

To mitigate the challenge of exponential sample complexity, we need to consider more stringent data coverage conditions. Early works on approximate dynamic programming (Munos, 2003, 2005, 2007) have offered guarantees on the suboptimality of the resulting policy, relying on assumptions related to good data coverage and mild distribution shift.

All-policy concentrability. The concept of data coverage has primarily been quantified using *concentration coefficients* (Munos, 2003). Given a data distribution μ, the concentration coefficient C is defined to be the smallest value such that, for any policy π, the following holds:

$$\max_{h \in [H-1]} \max_{s \in \mathcal{S}_h, a \in \mathcal{A}_h} \frac{\mathbb{P}^{\pi}(S_h = s, A_h = a)}{\mu_h(s, a)} \leq C.$$

If $\mu(s, a) = 0$ for some (s, a), we conventionally set $C = \infty$. In essence, the concentration coefficient serves to bound the ratio between the state-action distribution induced by any policy and the data distribution μ.

Several results have established bounds on the suboptimality of Approximate Policy Iteration (API) and Approximate Value Iteration (AVI) algorithms in terms of the concentration coefficient (Chen & Jiang, 2019; Farahmand et al., 2010; Munos, 2003, 2007). For example, it has been demonstrated that FQI outputs a near-optimal policy when C is small and the value function class satisfies the Bellman completeness condition. In such cases, the sample complexity scales linearly with C.

It's important to emphasize that the data coverage not only depends on the data distribution μ, but also on the underlying environment dynamics (implicitly in $\mathbb{P}^\pi$). The data coverage condition is designed to ensure that the data sufficiently covers all parts of the environment that any policy can encounter. This comprehensive coverage is crucial for obtaining reliable and meaningful results in the offline setting.

Various measures of data coverage have been introduced. For instance, M. Yin et al. (2021) assume the visitation distribution of the least occupied state-action pair is greater than or equal to d_m. This leads to corresponding sample complexity results that are on the order of $O(1/d_m)$. In a similar vein, Xie et al. (2021) propose a measure that takes into account the function class $\mathcal{F}$:

$$\max_{h\in[H-1]} \max_{f\in\mathcal{F}} \frac{\|f-\mathcal{T}f\|^2_{2,\mathbb{P}^\pi_h}}{\|f-\mathcal{T}f\|^2_{2,\mu_h}} \leq C.$$

The definition captures potential generalization across different state-action pairs with function approximation.

However, for many real-world applications, achieving a small concentration coefficient can be a significant challenge. This is particularly true when the data collection policy is not well-randomized or lacks exploratory behavior, which is often the case in practical settings. In such scenarios, the concentration coefficient may become very large or even infinite due to the omission of some state-action pairs from the dataset.

Munos (2007) offers valuable insights into the size of the concentration coefficient. To illustrate, consider a situation where the data distribution follows a uniform distribution (e.g., $\mu(s,a) = 1/|\mathcal{S}||\mathcal{A}|$), while the environment transition probabilities are less uniform. That is, certain policies may lead to visitation distribution that concentrates on a single state-action pair (e.g., $\mathbb{P}^\pi(S_h = s, A_h = a) = 1$ for some s, a and h). In such cases, the concentration coefficient can grow as large as the number of state-action pairs.

Coverage of an optimal policy. To mitigate the issue on strong assumptions about the concentration coefficient, an alternative condition is when the data covers a near-optimal policy. The fundamental idea behind these methods

is to constrain the policy to choose actions that have sufficient data coverage, which is effective if the given data contains near-optimal action selection.

For example, algorithms like BCQ (Fujimoto et al., 2019) and BEAR (Kumar et al., 2019) follow this approach. They bootstrap values from actions a only if the probability $\pi_b(a|s)$ exceeds a certain threshold b. In other words, they restrict action selection to those with sufficient data coverage. MBS-QI (Y. Liu et al., 2020) takes this concept a step further by considering state-action probabilities. It chooses to bootstrap from state-action pairs (s, a) only if $\mu(s, a)$ is above a predefined threshold. The algorithm is modified from FQI by replacing the bootstrap value $q_h(s, a)$ by $\tilde{q}_h(s, a) := \mathbb{I}\{\mu(s, a) \geq b\}q_h(s, a)$ and the policy is greedy with respect to $\tilde{q}_h(s, a)$. That is, if a state-action pair lacks sufficient data coverage, its value is treated as zero. They show that MBS-QI outputs a near-optimal policy if $\mathbb{P}^{\pi^*}(\mu(S_h, A_h) < b)$ is small for all $h \in [H-1]$. In other words, this method thrives when the data provides sufficient coverage for state-action pairs visited under an optimal policy π^*.

Besides Y. Liu et al. (2020), other studies consider single-policy concentrability (Rashidinejad et al., 2021; Zhan et al., 2022). This means that they only assume the ratio between the state-action distribution induced by an optimal policy π^* and the data distribution μ is bounded by C. Rashidinejad et al. (2021) demonstrate that a variant of value iteration with a pessimistic penalty can find a near-optimal policy under single-policy concentrability in the tabular case.

Empirically, these methods that constrain the policy to choose actions that have sufficient data coverage can be categorized into to two primary directions. The first direction focuses on constraining the divergence between the behavior policy and the output policy during the policy improvement step, primarily applicable to API algorithms. These constraints can be imposed either as direct policy constraints or through a penalty term added to the value function (Levine et al., 2020; Wu et al., 2019). Another approach involves constraining the policy set, ensuring that it only selects actions or state-action pairs with sufficient data coverage when updates are applied. This direction is primarily associated with AVI algorithms (Kumar et al., 2019; Y. Liu et al., 2020).

Another notable direction involves the adoption of pessimistic values for unknown state-action pairs, to encourage the agent to learn an improved policy that stays within the data-covered regions. This approach is exemplified by various algorithms: CQL (Kumar et al., 2020) penalizes the values for out-of-distribution actions and effectively learns a lower bound of the value estimates. A related idea is to constrain the bootstrap target to avoid out-of-distribution actions, introduced first in the BCQ algorithm (Fujimoto et al., 2019) with practical improvements given by IQL (Kostrikov et al., 2022). MOReL (Kidambi et al., 2020) learns a model and an unknown state-action detector to partition states, similar to the R-Max algorithm (Brafman & Tennenholtz, 2002). It adopts the principle of pessimism for these unknown states rather than optimism. Safe policy improvement methods (Laroche et al., 2019; P. Thomas et al., 2015a) rely on a high-confidence lower bound on the output policy performance, performing policy improvement only when the performance exceeds than a predefined threshold.

In practice, the results of pessimistic approaches are mixed. Some have been shown to be effective on the D4RL dataset (Fu et al., 2020). Other results, however, show methods can be too conservative and fail drastically when the behavior policy is not near-optimal (Kumar et al., 2020; Y. Liu et al., 2020). Further, the practical implementation of these methods can present challenges. Some of these methods require an estimate of the behavior policy or data distribution (Kumar et al., 2019; Y. Liu et al., 2020). Additionally, tuning hyperparameters for these methods in the offline setting is a non-trivial task, a topic that will be explored further in Chapter 4 of this thesis.

Though potentially less stringent than having a small concentration coefficient, this condition that the data covers an optimal policy can, in many practical scenarios, be challenging to meet. In simulated environments, such as Atari or Mujoco, it may be feasible to satisfy this assumption by using an online RL algorithm that can learn a good policy to collect optimal data. However, in real-world applications, the use case of offline RL is often driven by the fact that optimal policies are unknown and we don't have a simulator. In fact, one of the primary purposes of using offline RL is to get (significantly)

improved policies, which highlights the absence of prior knowledge about optimal behavior. It is also hard to carefully design a data collection policy to cover an unknown optimal policy, making it difficult to even check whether this assumption holds.

3.2.3 Function approximation conditions

Beyond data coverage, there is also an essential requirement on the function class. Recall that we have previously discussed the Bellman completeness condition. This condition, along with a standard data coverage condition, is sufficient to establish performance guarantees (Chen & Jiang, 2019).

These results naturally extend when the Bellman completeness condition holds approximately. In other words, the condition can be expressed as:

$$\sup_{q \in \mathcal{F}} \inf_{f \in \mathcal{F}} \|f - \mathcal{T}q\|_{\mu,p}^{p} \leq \varepsilon \tag{3.1}$$

for some $p \geq 1$ and $\varepsilon > 0$. The quantity on the left-hand side is called the inherent (one-step) Bellman error (Antos et al., 2008).

Theoretically, the condition can be satisfied under certain scenarios about the environment transition and the reward function. Fan et al. (2020) show that if the reward function and the transition probability are sufficiently smooth, then the $\mathcal{T}q$ belongs to a set of Hölder smooth functions. They further show that sparse ReLU networks can approximate the Hölder smooth functions accurately. This finding highlights that the approximate Bellman completeness condition can be satisfied even when neural networks are used as the function approximation. Additionally, the Bellman completeness condition is also satisfied in linear MDPs (Jin et al., 2020). In linear MDPs, both the reward function and transition probability can be represented as linear functions of a known feature map.

Empirically, Chang et al. (2022) present a method for learning a linear Bellman complete representation. Once the representation is learned, they perform offline policy evaluation with linear function approximation on top of these representations. However, their experimental results reveal an intriguing

finding. It appears that, in practice, FQE with a neural network still outperforms the proposed two-stage method. This outcome suggests that neural networks already offer sufficiently strong approximation capabilities, resulting in a low inherent Bellman error and consequently better performance.

Assuming the approximate Bellman completeness condition holds and the function class has finite element, the performance loss is often takes on the form of $O(\sqrt{\varepsilon} + \sqrt{\log |\mathcal{F}|/n})$. In this expression, the first term is the inherent Bellman error (approximation error) and the second term denotes the statistical complexity of the function class (estimation error). When we expand the size of the function class, the second term increases but it is possible that the first term decreases. This raises the important question of choosing a function class that balances between the approximation error and the estimation error. This motivates the exploration of model selection in the subsequent chapter.

As previously discussed in the background section, our focus lies on approximate dynamic programming methods such as FQE and FQI that rely on the Bellman completeness condition. However, there exist other approaches aimed at estimating quantities beyond the Bellman operator, including the density ratio (Xie et al., 2019) and the models of the MDP (Mannor et al., 2007; Uehara & Sun, 2022). Consequently, we require realizability conditions for the quantities under estimation.

Chapter 4

Towards Sample Efficient Offline Policy Selection

Offline reinforcement learning algorithms often require careful hyperparameter tuning. Consequently, before deployment, we need to select amongst a set of candidate policies. As yet, however, there is little understanding about the fundamental limits of this offline policy selection (OPS) problem. In this chapter we aim to provide clarity on when sample efficient OPS is possible, primarily by connecting OPS and off-policy policy evaluation (OPE). We first show a harndess result, that in the worst case, OPS is just as hard as OPE, by proving a reduction of OPE to OPS. We show that a simple OPE method, importance sampling, achieves a nearly minimax sample complexity. As a result, no OPS method can be more sample efficient than OPE in the worst case. Then we propose a Bellman error estimation method for OPS, and theoretically analyze when this method is sample efficient. We highlight that using BE generally has more requirements, but if satisfied, has an easy method for selecting its own hyperparameters and may be more sample efficient than OPE. We conclude by showing the difficulty of OPS for an offline Atari benchmark, and an empirical study comparing OPE and BE estimation.

4.1 Introduction

Offline RL is useful for many real-world applications, where learning from online interaction may be expensive or dangerous (Levine et al., 2020). There

have been significant advances in offline policy learning algorithms with demonstrations that performant policies can be learned purely from offline data (R. Agarwal et al., 2020). Successful use of these algorithms, however, requires careful hyperparameter selection (Gulcehre et al., 2020; Kumar et al., 2021; Wu et al., 2019).

Despite the fact that it is essential, hyperparameter selection for offline policy learning algorithms remains relatively unexplored. One of the reasons is that it is inherently hard, as we formalize in this chapter. Unlike supervised learning where we can use validation performance, in offline RL it is hard to get an accurate measure of policy performance. In fact, it is well known that off-policy policy evaluation (OPE)—estimating the performance of a policy from offline data—is hard (R. Wang et al., 2021). To emphasize this fact, in Figure 4.1, we visualize the poor correlation between the true performance and performance estimates using a standard OPE approach called Fitted Q Evaluation (FQE) (Le et al., 2019) . The datasets are not adversarially designed; rather they are two Atari datasets that were designed to provide reasonable data for offline learning algorithms.

Consequently, most offline RL papers sidestep the issue of hyperparameter selection and show results for ideal performance. In other words, they tune their hyperparameters by checking the performance of the policy in the environment. Consider, for example, the Conservative Q-learning (CQL) algorithm (Kumar et al., 2020), used to obtain the policies for Figure 4.1. For their experiments, CQL is built on top of already-tuned online algorithms for their environments, which implicitly relies on tuning using the environment rather than just the offline dataset. They then also directly tune the stepsize using the performance of the offline learned policy in the environment. Several papers (Kostrikov et al., 2022; Kumar et al., 2019) use the same methodology. Some works focus on designing algorithms that are robust to a specific hyperparameter (Cheng et al., 2022), or require only a small number of hyperparameters to be tuned (Fujimoto & Gu, 2021). These approaches, however, still require some hyperparameters to be tuned; they were tuned using the performance in the environment.

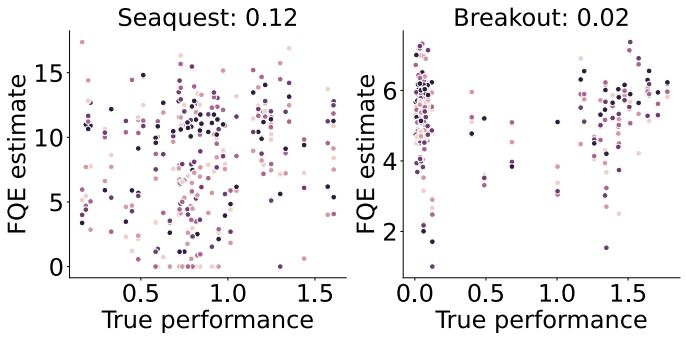


Figure 4.1: Correlation between true performance and estimated performance on two Atari datasets. We learn 90 policies offline using an algorithm called CQL (Kumar et al., 2020) and evaluate these policies using FQE on 5 different random datasets. These policies are generated by running CQL with a variety of different choices for two hyperparameters: the number of training steps and the conservative parameter. Each point in the scatter plot corresponds to a (policy, evaluation dataset) pair. The x-axis is the actual policy performance for that policy and y-axis is the estimated policy performance for that policy using that evaluation dataset. Colors represent different random seeds, namely different evaluation datasets. The Kendall rank correlation coefficient is shown in the label of the plot. If the FQE estimates accurately rank policies, we expect to see a strong linear relationship and a Kendall rank correlation coefficient close to 1. Neither of these behaviors are seen here, and it is clear FQE does not provide an effective mechanism to rank policies.

There are, however, a handful of works that use only the offline dataset to select hyperparameters. These can be categorized as papers focused on algorithms for hyperparameter selection in offline RL (Paine et al., 2020; Tang & Wiens, 2021; M. Yang et al., 2022), papers that provide theoretical algorithms in specialized settings (Farahmand & Szepesvári, 2011; J. Lee et al., 2022), or papers that introduce offline RL algorithms and use sensible heuristics to select hyperparameters in their experiments (Kumar et al., 2021; Qi et al., 2022; Trabucco et al., 2021; T. Yu et al., 2021). For the first category, these algorithms are all based on OPE. For the second category, these algorithms are applicable only in specialized settings and require strong conditions to obtain the provided theoretical guarantees. For the last category, the heuristics are often tailored to the specific algorithm presented in these papers and do not come with performance guarantees. For example, Kumar et al. (2021) used statistics about the TD errors to select the number of training steps for CQL.

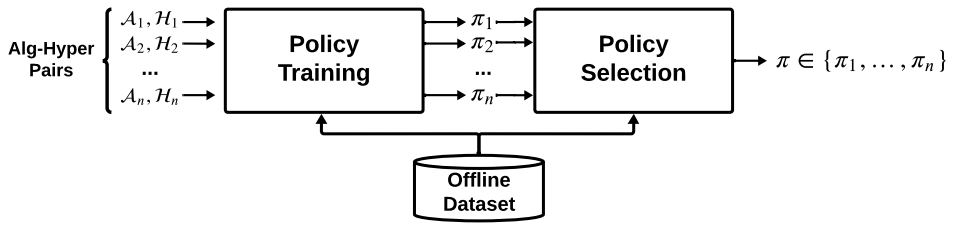


Figure 4.2: The offline RL pipeline with OPS. In the policy training phase, n algorithm-hyperparameter pairs are trained on an offline dataset to produce n candidate policies. These n policies, combined again with offline data (potentially a validation dataset) are run with an OPS algorithm to find a final policy amongst the candidate policies.

In general, there are as yet many open questions about the feasibility of hyperparameter selection in offline RL, both in theory and in practice. This is doubly true if we go beyond hyperparameter selection and consider selecting amongst different policy learning algorithms as well, each with their own hyperparameters. In this chapter, we consider this more general problem, called *offline policy selection* (OPS), that can be used to select policy learning algorithms and their hyperparameters. Figure 4.2 shows the full offline RL pipeline with OPS.

As mentioned above, OPE is a general approach to OPS, but we want to understand *when, or even if, alternative approaches can outperform OPE for OPS*. OPE might not be the ideal answer to OPS. One reason is that OPE requires a large number of samples to evaluate any given policy in the worst case (R. Wang et al., 2021). We might intuit that OPS may, at least in some cases, be easier than OPE, since OPS need only identify the best-performing policy, whereas OPE aims to estimate each policy's value accurately. For example, Doroudi et al. (2017) designed a specialized importance sampling (IS) estimator to compare two policies, to remove a bias towards policies that result in shorter trajectories, improving on the standard IS estimator used in OPE. Therefore, the first question that needs to be answered is: *Is OPS easier than OPE?* Surprisingly, this basic question has not yet been explicitly answered.

Another key reason current OPE estimators do not resolve the OPS prob-

lem is because these OPE methods themselves have hyperparameters. For example, the MAGIC estimator (P. Thomas & Brunskill, 2016) and the clipped importance sampling (IS) estimator (Bottou et al., 2013) are sensitive to their hyperparameters. Even a variant of the IS estimators designed specifically for OPS (M. Yang et al., 2022) has hyperparameters. Other methods like FQE require learning a value function, and how to select the function class is still an open problem.

The natural alternative to OPE is to use Bellman errors (BE), but there is mixed evidence on whether such approaches are effective. Tang and Wiens (2021) empirically show that selecting the candidate value function with the smallest TD errors perform poorly because the value functions are often overestimated; they conclude OPE is necessary. Similarly, Paine et al. (2020) present positive experimental results using FQE—an OPE approach—for OPS but concluded that the use of TD errors was ineffective. Other works, however, have developed new OPS algorithms that rely on the use of BE. Zhang and Jiang (2021) propose a value-function selection algorithm called BVFT, which computes the (empirical) projected BE for each pair of candidate value functions. J. N. Lee et al. (2022) provide a method for selecting the best function class from a nested set of function classes for Fitted Q-Iteration. These theoretically-sound methods, however, either rely on strong assumptions or are applicable only in specialized settings. It is unclear whether these methods are better than OPE.

Contributions. In this chapter, we make contributions towards answering the above question. Our first contribution is to prove that OPS inherits the same hardness results as OPE, which has never been formally shown in the literature. We show that the sample complexity of the OPS problem is lower-bounded by the samples needed to perform OPE. We further show that an OPS algorithm that simply chooses the policy with the highest IS estimate achieves a nearly minimax sample complexity, which is exponential in the horizon. This result implies no OPS approach can avoid the exponential sample complexity in the worst case, and we must consider additional assumptions to enable

sample efficient OPS.

We then investigate the utility of using BE for OPS. We propose a simple BE algorithm, called Identifiable BE selection (IBES), with a simple way to select its own hyperparameters. This is contrast to many OPE methods, even simpler approaches like FQE, for which it can be hard to select hyperparameters. We show that IBES can provide improved sample efficiency, but under stricter requirements on data coverage and on the candidate set compared to FQE. In an empirical study, we systematically compare different BE methods with varying sample sizes and show IBES is consistently more sample efficient across multiple environments.

From our empirical study, we also show the inherent difficulty of OPS in a larger scale experiment in Atari benchmark. Notably, we show that all the OPS methods suffer relatively high regret compared to using the tuned hyperparameters, in many cases performing even worse than random hyperparameter selection. Recall that the tuned hyperparameters are obtained by using performance in the real environment, which is not feasible in practice but nonetheless commonplace in offline RL experiments. This result in Atari highlights that many results in the literature report unrealistically favorable results for their offline algorithms, because they show results for these tuned hyperparameters rather than hyperparameters chosen using only the offline dataset.

4.2 On the sample complexity of OPS

We consider the offline policy selection (OPS) problem and off-policy policy evaluation (OPE) problem. We follow a similar notation and formulation used in Xiao et al. (2022) to formally describe these problem settings. The OPS problem for a fixed number of episodes n is given by the tuple $(\mathcal{S}, \mathcal{A}, H, \nu, n, \mathcal{I})$. $\mathcal{I}$ is a set of instances of the form (M, d_b, Π) where $M \in \mathcal{M}(\mathcal{S}, \mathcal{A}, H, \nu)$ specifies an MDP with state space $\mathcal{S}$, action space $\mathcal{A}$, horizon H and the initial state distribution ν, d_b is a distribution over a trajectory $(S_0, A_0, R_0, \ldots, R_{H-1})$ by running the behavior policy π_b on M, and Π is a finite set of candidate policies.

We consider the setting where Π has a small size and does not depend on S, A or H.

An OPS algorithm takes as input a batch of data D, which contains n trajectories, and a set of candidate policies Π, and outputs a policy $\pi \in \Pi$. The aim is to find OPS algorithms that are guaranteed to return the best candidacy policy with high probability.

Definition 1 ((ε, δ)-sound OPS algorithm)**.** Given $\varepsilon > 0$ and $\delta \in (0, 1)$, an OPS algorithm $\mathcal{L}$ is (ε, δ)-sound on instance (M, d_b, Π) if

$$\Pr_{D \sim d_b} \left(J_M(\mathcal{L}(D, \Pi)) \geq J_M(\pi^\dagger) - \varepsilon \right) \geq 1 - \delta,$$

where $\pi^\dagger$ is the best policy in Π. We say an OPS algorithm $\mathcal{L}$ is (ε, δ)-sound on the problem $(\mathcal{S}, \mathcal{A}, H, \nu, n, \mathcal{I})$ if it is sound on any instance $(M, d_b, \Pi) \in \mathcal{I}$.

Given a pair (ε, δ), the sample complexity of OPS is the smallest integer n such that there exists a behavior policy π_b and an OPS algorithm $\mathcal{L}$ such that $\mathcal{L}$ is (ε, δ)-sound on the OPS problem $(\mathcal{S}, \mathcal{A}, H, \nu, n, \mathcal{I}(\pi_b))$ where $\mathcal{I}(\pi_b)$ denotes the set of instances with data distribution d_b. That is, if the sample complexity is lower-bounded by a number N_{OPS}, then, for any behavior policy π_b, there exists an MDP M and a set of candidate policies Π such that any (ε, δ)-sound OPS algorithm on (M, d_b, Π) requires at least N_{OPS} episodes.

Similarly, the OPE problem for a fixed number of episodes n is given by $(\mathcal{S}, \mathcal{A}, H, \nu, n, \mathcal{I})$. $\mathcal{I}$ is a set of instances of the form (M, d_b, π) where M and d_b are defined as above, and π is a target policy. An OPE algorithm takes as input a batch of data D and a target policy π, and outputs an estimate of the policy value. The goal is to estimate the true value accurately.

Definition 2 ((ε, δ)-sound OPE algorithm)**.** Given $\varepsilon \in (0, V_{max}/2)$ and $\delta \in (0, 1)$, an OPE algorithm $\mathcal{L}$ is (ε, δ)-sound on instance (M, d_b, π) if

$$\Pr_{D \sim d_b} \left(|\mathcal{L}(D, \pi) - J_M(\pi)| \leq \varepsilon \right) \geq 1 - \delta.$$

We say an OPE algorithm $\mathcal{L}$ is (ε, δ)-sound on the problem $(\mathcal{S}, \mathcal{A}, H, \nu, n, \mathcal{I})$ if it is sound on any instance $(M, d_b, \pi) \in \mathcal{I}$.

Note that ε should be less than $V_{max}/2$ otherwise the bound is trivial.

4.2.1 OPE as subroutine for OPS

It is obvious that a sound OPE algorithm can be used for OPS, since we can run the sound OPE algorithm to evaluate each candidate policy and select the policy with the highest estimate. As a result, the sample complexity of OPS is upper-bounded by the sample complexity of OPE up to a logarithmic factor. For completeness, we state this formally in the theorem below.

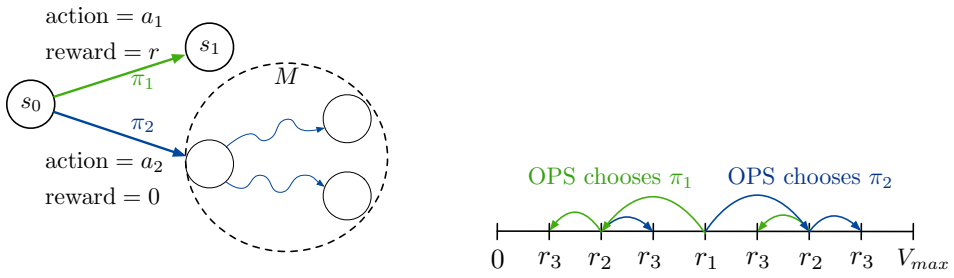
Theorem 1 (Upper bound on sample complexity of OPS)**.** Given an MDP M, a data distribution d_b, and a set of policies Π, suppose that, for any pair (ε, δ), there exists an (ε, δ)-sound OPE algorithm $\mathcal{L}$ on any OPE instance $I \in \{(M, d_b, \pi) : \pi \in \Pi\}$ with a sample size at most $O(\mathrm{N}_{OPE}(S, A, H, 1/\varepsilon, 1/\delta))$. Then there exists an (ε, δ)-sound OPS algorithm for the OPS problem instance (M, d_b, Π) which requires at most $O(\mathrm{N}_{OPE}(S, A, H, 2/\varepsilon, |\Pi|/\delta))$ episodes.

In terms of the sample complexity, we have an extra $\sqrt{\log |\Pi|/n}$ term for OPS due to the union bound. For hyperparameter selection in practice, the size of the candidate set is often much smaller than n, so this extra term is negligible. However, if the set is too large, complexity regularization (Bartlett et al., 2002) may need to be considered. In this chapter, we only consider a finite candidate set.

4.2.2 OPS is not easier than OPE

We have shown that OPS is sample efficient when OPE is sample efficient. However, it remains unclear whether OPS can be sample efficient when OPE is not. In the following theorem, we lower bound the sample complexity of OPS by the sample complexity of OPE. As a result, both OPS and OPE suffer from the same hardness result.

Theorem 2 (Lower bound on sample complexity of OPS)**.** Suppose for any data distribution d_b and any pair (ε, δ) with $\varepsilon \in (0, V_{max}/2)$ and $\delta \in (0, 1)$, there exists an MDP M and a policy π such that any (ε, δ)-sound OPE algorithm requires at least $\Omega(\mathrm{N}_{OPE}(S, A, H, 1/\varepsilon, 1/\delta))$ episodes. Then there exists an MDP M' with $S' = S + 2$, $H' = H + 1$, and a set of candidate policies



(a) Constructing M' and a candidate policy set

(b) Searching for the true policy value

Figure 4.3: Visual depiction of the reduction of OPE to OPS. Given a MDP M and a target policy π, we can construct a new MDP M' and two candidate policies $\{\pi_1, \pi_2\}$ for OPS, as shown in (a). The MDP construction was first mentioned in R. Wang et al. (2021). π_1 chooses a_1 in s_0 and is otherwise arbitrary, π_2 chooses a_2 and is otherwise identical to the target policy π. Figure (b) describes the search procedure to find the policy value by calling the OPS subroutine. When the OPS query returns π_1, we follow the green arrow. When the OPS query returns π_2, we follow the blue arrow. We can keep searching for the true policy value by setting r for the OPS query, until the desired precision is reached.

such that for any pair (ε, δ) with $\varepsilon \in (0, V_{max}/3)$ and $\delta \in (0, 1/m)$ where $m := \lceil \log(V_{max}/\varepsilon) \rceil \geq 1$, any (ε, δ)-sound OPS algorithm also requires at least $\Omega(\mathrm{N}_{OPE}(S, A, H, 3/2\varepsilon, 1/m\delta))$ episodes.

The proof sketch is to construct an OPE algorithm that queries OPS as a subroutine, as demonstrated in Figure 4.3. As a result, the sample complexity of OPS is lower bounded by the sample complexity of OPE. The proof can be found in Appendix A.

There exist several hardness results for OPE in tabular settings and with linear function approximation (R. Wang et al., 2021; M. Yin & Wang, 2021). Theorem 2 implies that the same hardness results hold for OPS since lower bounds for OPE are also lower bounds for OPS. We should not expect to have a sound OPS algorithm without additional assumptions to make OPE work. Theorem 2, however, does not imply that OPS and OPE are always equally hard. There are instances where OPS is easy but OPE is not. For example, when all policies in the candidate set all have the same value, any random policy selection is sound. However, OPE can still be difficult in such cases.

4.3 Minimax sample complexity for finite horizon finite MDPs

In the previous section, we show that OPS and OPE problem are equally difficult in the worst case scenario. In this section, we show a lower bound on sample complexity of OPS for finite horizon finite MDPs, and that an OPE method (nearly) matches the lower bound.

We first present an exponential lower bound, using the lower bound construction from Xiao et al. (2022) and Theorem 2.

Corollary 1 (Lower bound on the sample complexity of OPS)**.** For any positive integers S, A, H with $S > 2H$ and a pair (ε, δ) with $0 < \varepsilon \leq \sqrt{1/8}$, $\delta \in (0, 1)$, any (ε, δ)-sound OPS algorithm needs at least $\tilde{\Omega}(A^{H-1}/\varepsilon^2)$ episodes.

We show that importance sampling (IS) achieves the lower bound. Recall the IS estimator (Rubinstein, 1981) is given by

$$\hat{J}(\pi) = \frac{1}{n} \sum_{i=1}^{n} \prod_{h=0}^{H-1} \pi(A_h^{(i)}|S_h^{(i)})/\pi_b(A_h^{(i)}|S_h^{(i)}) G_i$$

where $G_i = \sum_{h=0}^{H-1} R_h^{(i)}$ and n is the number of episodes in the dataset D.

Corollary 2 (Upper bound on the sample complexity of OPS)**.** Suppose the data collection policy is uniformly random, that is, $\pi_b(a|s) = 1/A$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$, and $|G_i| \leq V_{max}$ almost surely. Then the selection algorithm $\mathcal{L}$ that selects the policy with the highest IS estimate is (ε, δ)-sound with $O(A^H V_{max} \ln(|\Pi|/\delta)/\varepsilon^2)$ episodes.

Note that Y.-X. Wang et al. (2017) have shown that IS estimator achieved the minimax mean squared error for the OPE problem. Our result shows that IS also achieves a (nearly) minimax sample complexity for the OPS problem, which is different from their work.

These results suggest that IS achieves a nearly minimax optimal sample complexity for OPS up to a factor A and logarithmic factors. There are other improved variants of IS, including per-decision IS and weighted IS (Precup et

al., 2000; Sutton & Barto, 2018). None of these variants can help reduce sample complexity in the worst case because the lower bound in Corollary 1 holds for any OPS algorithm. This result suggests that we need to consider additional assumptions on the environment, the data distribution, or the candidate set to perform sample efficient OPS.

One direction is to consider when OPE can be sample efficient, since we can leverage Theorem 1 to inherit the sample complexity result from OPE to OPS. There is a wealth of literature on OPE, in particular, FQE and MIS (or DICE) methods have been shown to be effective for OPS empirically (Paine et al., 2020; M. Yang et al., 2022). FQE and MIS methods require a standard data coverage assumption, that is, we need data coverage for all the candidate policies. Data coverage is commonly measured by the concentration coefficient, first introduced in Munos (2007).[1] The precise definition will be provided in the next section.

Besides data coverage, we also need additional assumptions on the function approximation. For FQE, we need a function class $\mathcal{F}$ that is closed under $\mathcal{T}^\pi$ for all $\pi \in \Pi$, that is, $\mathcal{T}^\pi q \in \mathcal{F}$ for any $q \in \mathcal{F}$. Assume $\mathcal{F}$ has finitely many elements (for simplicity only), the result from Duan et al. (2021) can be extended to show that the sample complexity of using FQE for OPS is $O(H^4 \log(|\mathcal{F}||\Pi|H/\delta)/\varepsilon^2)$, under both assumptions on data coverage and function class. There are corresponding results for the MIS (or DICE) methods (Nachum et al., 2019; Uehara et al., 2020). See Table 4.1 for a brief summary.

In summary, we showed an exponential lower bound on the sample complexity for OPS, and we need to consider additional assumptions such as data coverage and function approximation that enable sample efficient OPE.

4.4 Bellman error selection for OPS

In this section, we shift our focus to the use of Bellman error (BE) for OPS. There is mixed evidence on whether such approaches are effective compared to

[1]There are other measures of data coverage (Xie et al., 2021; M. Yin & Wang, 2021). However, we focus on the concentration coefficient to clearly compare assumptions between different methods.

OPE. As a result, we first provide conditions when BE is useful for OPS, which allows us to clearly compare OPE and BE methods. We then propose a BE selection method that has a simple method to select its own hyperparameter, and proved the sample complexity of the method.

Suppose we are given a set of candidate value functions $\mathcal{Q} := \{q_1, \ldots, q_K\}$ and let $\Pi = \{\pi_1, \ldots, \pi_K\}$ be the set of corresponding greedy policies, a common strategy is to select the action value function with the smallest BE (Farahmand & Szepesvári, 2011). In the finite horizon setting, we define the Bellman error with respect to q_i as

$$\mathcal{E}(q_i) := \frac{1}{H} \sum_{h=0}^{H-1} \|q_i - \mathcal{T} q_i\|_{2,\mu_h}^2$$

where $\|q\|_{p,\mu_h} := (\sum_{(s,a)\in\mathcal{S}_h\times\mathcal{A}} \mu_h(s,a)|q(s,a)|^p)^{1/p}$. We define (ε, δ)-sound *BE selection* in the following.

Definition 3 ((ε, δ)-sound BE selection)**.** Given a set of candidate value functions $\mathcal{Q}$, $\varepsilon > 0$ and $\delta \in (0, 1)$, an BE selection algorithm $\mathcal{L}$, which takes $D, \mathcal{Q}$ as input and outputs $q \in \mathcal{Q}$, is (ε, δ)-sound on $\mathcal{Q}$ if

$$\mathcal{E}(\mathcal{L}(D, \mathcal{Q})) \leq \min_{i=1,\ldots,|\mathcal{Q}|} \mathcal{E}(q_i) + \varepsilon$$

with probability at least $1 - \delta$.

4.4.1 When is BE selection useful for OPS?

In order to relate BE selection to OPS, we need data coverage for both the candidate policies Π and an optimal policy (the first assumption in the following corollary). We present an error bound for OPS using BE selection.

Corollary 3 (Error amplification for OPS using BE selection)**.** Suppose

1. there exists a constant C such that $\forall \pi \in \Pi \cup \{\pi^*\}$, $\max_{h\in[H]} \max_{s\in\mathcal{S}_h, a\in\mathcal{A}_h} \frac{d_h^\pi(s,a)}{\mu_h(s,a)} \leq C$,

2. the suboptimality of the candidate set is small, that is, $\min_{q\in\mathcal{Q}} \mathcal{E}(q) \leq \varepsilon_{sub}$,

3. there exists an $(\varepsilon_{est}, \delta)$-sound BE selection algorithm $\mathcal{L}$ on $\mathcal{Q}$,

then the OPS algorithm outputs the greedy policy with respect to $\mathcal{L}(D, \mathcal{Q})$ is $(2H\sqrt{C(\varepsilon_{sub} + \varepsilon_{est})}, \delta)$-sound.

That is, even if we have a good BE selection algorithm with a small ε_{est}, the error for OPS is amplified by C and ε_{sub}. Compared to OPE methods such as FQE, BE has an additional error ε_{sub} which does not go to zero even when we can collect more samples for OPS. Only ε_{est} goes to zero as n goes to infinity, which will be discussed in the next subsection.

To see how poor the guarantee can be due to ε_{sub}, suppose we have two action values $q_1 = 100q^*$ and $q_2 = q^* +$some random noise, then q_1 has a large Bellman error but π_1 is actually optimal. We would choose q_2 since it has a lower Bellman error, even when we can collect an infinite number of samples. To obtain a meaningful guarantee, we need to include another value function, for example, $q_3 = q^*$ to make $\varepsilon_{sub} = 0$. As we collect more samples, we can estimate the Bellman error more accurately and eventually choose q_3.

At the same time, as we get more offline data, then we might actually change our candidate set. If we have more samples for training offline algorithms and generating the candidate set, it is more likely that there is one action value function in the candidate set that is close to optimal. That is, for a fixed candidate set, ε_{sub} does not get smaller as we collect more samples for OPS, but if we use those offline samples to train the candidates policies, then ε_{sub} might get smaller.

4.4.2 A sample-efficient and hyperparameter-free BE selection method

In this subsection, we propose a sound BE selection method, which we call Identifiable BE Selection (IBES). In deterministic environments, the BE can be easily estimated using TD errors. Given a state-action value q with $v_q(s) := \max_a q(s, a)$ the corresponding state value, the BE estimate is $\text{TDE}(q) := \frac{1}{|D|}\sum_{(s,a,s',r)\in D}(q(s, a) - r - v_q(s'))^2$.

In stochastic environments, estimating BE typically involves an additional regression problem (Antos et al., 2008). Antos et al. (2008) propose an esti-

mator for the BE by introducing an auxiliary function $g \in \mathcal{G}$:

$$\begin{aligned}\hat{\mathcal{E}}(q) &= \tfrac{1}{|D|} \sum_{(s,a,s',r)\in D} (q(s,a) - r - v_q(s'))^2 - \min_{g\in\mathcal{G}} \tfrac{1}{|D|} \sum_{(s,a,s',r)\in D} (g(s,a) - r - v_q(s'))^2 \\ &= \max_{g\in\mathcal{G}} \left[\tfrac{1}{|D|} \sum_{(s,a,s',r)\in D} (q(s,a) - r - v_q(s'))^2 - \tfrac{1}{|D|} \sum_{(s,a,s',r)\in D} (g(s,a) - r - v_q(s'))^2 \right].\end{aligned} \tag{4.1}$$

The sample complexity of estimating the BE depends on the complexity of the function class $\mathcal{G}$, as shown in the theorem below.

Theorem 3. Suppose all $q \in \mathcal{Q}$ and $g \in \mathcal{G}$ take value in $[0, V_{max}]$. Let $q^\dagger$ be the action value function with the smallest estimated Bellman error $\hat{\mathcal{E}}(q)$. Then with probability at least $1-\delta$, for some constant c_0 (ignoring approximation and optimization error),

$$\mathcal{E}(q^\dagger) \leq \min_{i=1,\ldots,|\mathcal{Q}|} \mathcal{E}(q_i) + c_0 V_{max}^2 \sqrt{\log(2|\Pi|H/\delta)/n} + c_0 V_{max} \mathcal{R}_n^\mu(\mathcal{G})$$

where $\mathcal{R}_n^\mu(\mathcal{G})$ is the Rademacher complexity of the function class $\mathcal{G}$.

Assume $\mathcal{G}$ has finite elements (for simplicity only), the sample complexity of finding a good action value function is $O(H^4 \log(|\mathcal{G}||\Pi|H/\delta)/\varepsilon_{est}^2)$. Compared to deterministic environments, we have an additional term that depends on the size of the function class. The proof can be found in Appendix A.

We can rewrite the inner term in Equation (4.1) by the change of variable (Dai et al., 2018; Patterson et al., 2022). Let $t(r, s') := r + v_q(s')$ be the target and $\delta(s, a, r, s') := t(r, s') - q(s, a)$ be the TD error to simplify the equations. Consider a new auxiliary function $h(s, a) = g(s, a) - q(s, a)$, then the inner term in Equation (4.1) is

$$\begin{aligned}&\tfrac{1}{|D|} \sum_{(s,a,s',r)\in D} \left[(q(s,a) - r - v_q(s'))^2 - (g(s,a) - r - v_q(s'))^2 \right] \\ &= \tfrac{1}{|D|} \sum_{(s,a,s',r)\in D} \left[(t(r,s') - q(s,a))^2 - (t(r,s') - g(s,a))^2 \right] \\ &= \tfrac{1}{|D|} \sum_{(s,a,s',r)\in D} \left[2h(s,a)\delta(s,a,r,s') - h(s,a)^2 \right].\end{aligned} \tag{4.2}$$

Algorithm 3 Identifiable BE Selection (IBES) with holdout validation

Input: Candidate set $\mathcal{Q}$, training data D, validation data D_{val}, a set of function classes $\mathcal{G}_1, \ldots, \mathcal{G}_M$ (for model selection)
Let $\delta(s, a, r, s') := r + v_q(s') - q(s, a)$
for $q \in \mathcal{Q}$ **do**
  for $m = 1, \ldots, M$ **do**
    Perform regression: $\hat{g}_m \leftarrow \min_{g \in \mathcal{G}_m} \frac{1}{|D|} \sum_D (g(s, a) - \delta(s, a, r, s'))^2$
    Compute validation error: $l(\hat{g}_m) \leftarrow \frac{1}{|D_{val}|} \sum_{D_{val}} (\hat{g}_m(s, a) - \delta(s, a, r, s'))^2$
  Find the best function class: $k \leftarrow \arg\min_{m=1,\ldots,M} l(\hat{g}_m)$
  Estimate the Bellman error for q: $\mathrm{BE}(q) \leftarrow \frac{1}{|D|} \sum_D 2\hat{g}_k(s, a)\delta(s, a, r, s') - \hat{g}_k(s, a)^2$
Output: $q^\dagger \leftarrow \arg\min_{q \in \mathcal{Q}} \mathrm{BE}(q)$

That is, we can also use an auxiliary function h to predict $\mathcal{T}q - q$, instead of $\mathcal{T}q$, and use the auxiliary function to estimate the Bellman error. The benefit is that the Bellman errors are more likely to be predictable. Under the conditions for Corollary 3, if there is an action-value function $q \in \mathcal{Q}$ with a small BE, the Bellman errors are nearly zero everywhere and any reasonable function class are able to represent the solution. The remaining part is to find a function class that has low approximation error and a low statistical complexity.

Fortunately, we can perform model selection to choose the function approximation $\mathcal{G}$. This is because we are running regression with fixed targets in Eq (4.1), and model selection for regression is well-studied. For example, consider a finite set of potential function classes $\mathcal{G}_1, \ldots, \mathcal{G}_M$, we can use a hold-out validation set to select the best function class and other hyperparameters. Therefore, we can choose a function class such that it has a low approximation error and a small complexity measure, which can potentially result in improved sample efficiency. We describe the full procedure of IBES in Algorithm 3.

Note that we can also directly use $\frac{1}{|D|} \sum_{(s,a)} h(s, a)^2$ as an estimate for the Bellman error. In our experiment, we found that it performs similar as using Eq (4.2).

4.4.3 Comparison to existing methods

There are other works consider selecting a value function that has the smallest BE or is the closest to the optimal value function. Farahmand and Szepesvári (2011) consider selecting a value function such that, with high probability, the output value functions has the smallest BE. They propose to fit a regression model $\tilde{q}_i$ to predict $\mathcal{T}q_i$ and bound the BE by $\|q_i - \tilde{q}_i\|_{2,\mu}^2 + b_i$ where the first term can be viewed as the (empirical) projected Bellman error, and the second term b_i is a high-probability upper bound on the excess risk of the regression model, which is assumed to be given. There is a concurrent work on using BE selection method for OPS (Zitovsky et al., 2023). They propose a method called Supervised Bellman Validation (SBV), which is essentially an empirical version of the method from Farahmand and Szepesvári (2011), without an additional upper bound on the excess risk. They did not provide any sample complexity guarantee. In our experiments, we find that our method outperforms SBV in terms of sample efficiency, likely because the auxiliary function is used to predict the Bellman error, instead of $\mathcal{T}q$.

Zhang and Jiang (2021) propose to use a (empirical) projected Bellman error, called BVFT loss, with piecewise constant function classes. Their selection algorithm chooses the value function with the smallest BVFT loss, assuming q^* is in the candidate set (approximately) and a stronger data assumption is satisfied. Interestingly, this condition on having q^* is essentially equivalent to our condition requiring small ε_{sub}, since q^* has exactly zero BE. The algorithms, though, are quite different from our work. Their method is computationally expensive since it scales with $O(|\Pi|^2)$ instead of $O(|\Pi|)$, making the method impractical when the candidate set is large. Our method requires a weaker data coverage assumption and the computation cost scales linearly with $O(|\Pi|)$. Note that the sample complexity of our methods depends on the function class that is used to perform the regression. BVFT can be viewed as using the piecewise constant function classes to fit the Bellman target, the sample complexity depends on the piecewise constant function classes, which is measured by the number of discretization bins $(V_{max}/\varepsilon_{dct})^2$ and ε_{dct} is the

Method	Assumptions	Sample Complexity
IS	$\pi_b(a\|s) > 0$ if $\pi(a\|s) > 0$ for some $\pi \in \Pi$	$O(A^H H \ln(\|\Pi\|/\delta)/\varepsilon^2)$
FQE	(1) Data coverage for Π (2) $\mathcal{F}$ is closed under $\mathcal{T}^\pi$ for all $\pi \in \Pi$	$O(H^4 \log(\|\mathcal{F}\|\|\Pi\|H/\delta)/\varepsilon^2)$
IBES	(1) Data coverage for $\Pi \cup \{\pi^*\}$ (2) Small ε_{sub} (3) $\mathcal{G}$ realizes $(\mathcal{T}q - q)$ for $q \in \mathcal{Q}$	$O(H^4 \log(\|\mathcal{F}\|\|\Pi\|H/\delta)/\varepsilon_{est}^2)$
BVFT	(1) Stronger data coverage (2) $q^* \in \mathcal{Q}$ (which implies small ε_{sub})	Number of partitions

Table 4.1: A summary of OPS methods. These sample complexity results are derived in this chapter, except for the sample complexity of BVFT given in Zhang and Jiang (2021). The first two methods are OPE methods and the last two methods use action value functions. The assumptions for function approximation can be relaxed to hold approximately.

discretization resolution.

4.4.4 Should we use BE selection or OPE?

Table 4.1 summarizes the comparison of BE and OPE methods. OPS using IBES and BVFT requires stronger data coverage assumptions since it needs coverage not only for the candidates policies, but also π^*. Moreover, the guarantee for OPS using IBES and BVFT can be poor due to the additional term ε_{sub}, which does not decrease as we collect more samples for OPS. We need at least one of the action-value functions to be close to the optimal action-value function. For OPE, we can perform OPS even if none of the candidate policies are close to optimal. Therefore, OPE is a more robust and reliable method for OPS.

On the other hand, if we satisfy this stronger data coverage condition and have small ε_{sub}, then IBES has several advantages. IBES can be much more sample efficient in deterministic environments, or even in stochastic environments by choosing an appropriate function approximation. More importantly, for IBES, we are not plagued by the issue of having hard-to-specify hyperparameters. This is critical for the offline setting, where we cannot test different hyperparameter choices in the environment.

4.5 Experimental results

In this section, we empirically investigate the different BE methods as well as FQE for OPS. The goal in the experiments is to gain a better understanding of how IBES and FQE perform when we vary important factors such as data coverage, sample size, and candidate policies. We first compare different BE-based methods to show the advantage of our proposed method IBES, and investigate the differences between IBES and FQE in classic control environments where we vary the data coverage. Finally, we perform an experiment on the Atari offline benchmark dataset.

To evaluate the performance of OPS, we consider the normalized top-k regret used in Zhang and Jiang (2021). Top-k regret is the gap between the best policy within the top-k policies, and the best policy among all candidate policies. We then normalized the regret by the difference between the best and the worst policy, so the normalized regret is between 0 and 1. The normalized regret can be interpreted as the percentage of degradation if we use offline selection instead of online selection, since online selection with Monte Carlo evaluation can always achieve a regret of zero. OPS corresponds to $k = 1$; for most results we use $k = 1$, but include some results for $k > 1$. All hyperparameters for OPS methods are selected using only the datasets, not by peaking at performance on the real environment; for more details, see Appendix A.2.

4.5.1 Comparison between BE-based methods

In the first set of experiments, we compare different BE-based methods for OPS. We conduct experiments on two standard RL environments: Acrobot and Cartpole. We also include the stochastic versions of these environments with sticky actions (Machado et al., 2018), which we call Stochastic Acrobot and Stochastic Cartpole. We generate a set of candidate policy-value pairs by running CQL with different hyperparameters on a batch of data collected by a trained policy with random actions taken 40% of the time, and generate datasets for OPS that provide good data coverage. We then use either FQE or IBES for OPS. We also included SBV (Zitovsky et al., 2023) and BVFT

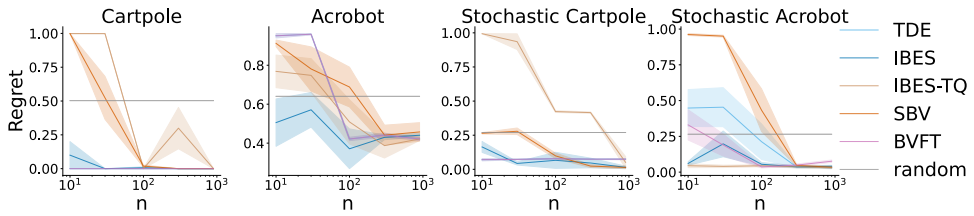


Figure 4.4: Comparison between BE methods. The figure shows the normalized top-1 regret with varying sample size, averaged over 10 runs with one standard error. IBES consistently achieves the lowest regret across environments.

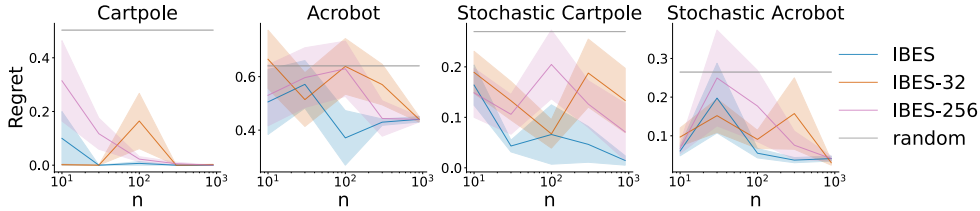


Figure 4.5: Comparison to BE with a fixed number of hidden units. The figure shows the normalized top-1 regret with varying sample size, averaged over 10 runs with one standard error. IBES with model selection consistently achieves the lowest regret across environments.

(Zhang & Jiang, 2021),[2] and a random selection baseline which selects a policy uniformly at random from the candidate set.

We first compare different BE methods, and investigate the effect of using the Bellman error $\mathcal{T}q - q$ as target. We include a baseline that use $\mathcal{T}q$ as target, called IBES-TQ. Note that we also perform model selection for SBV and IBES-TQ, similar to IBES. In Figure 4.4, we can see all BE methods converge to the same performance as sample size gets larger, but IBES using the Bellman error as target is much more sample efficient than IBES-TQ and SBV across all environment. This is likely due to the fact that Bellman error is easier to predict so using a smaller neural network is sufficient and has a better sample efficiency. We also found BVFT performs similar to TDE (which are overlapping in Cartpole, Acrobot and Stochastic Cartpole), which is likely due to the fact that BVFT with small discretization is equivalent to TDE.

[2]We modified the BVFT implementation from the author of Zhang and Jiang (2021) (https://github.com/jasonzhang929/BVFT_empirical_experiments/).

Now we show that the model selection procedure is important for IBES, by comparing to IBES with a fixed number of hidden units. In our experiment, we use a two layer neural network model as the function approximation, and perform model selection to find the number of hidden units from the set $\{32, 64, 128, 256\}$ for IBES. In Figure 4.5, we include two baselines called IBES-32 and IBES-256, which has the hidden units of size 32 and 256 respectively. In general, IBES with model selection achieves regret less than or equal to IBES with a fixed number of hidden units across all environments. The performance of IBES with model selection and IBES-256 converges as the sample size gets larger. This result shows an improvement in terms of sample efficiency due to model selection. In stochastic environments, IBES with a small number of hidden units does not work well even with a large sample size. The results suggest that the ability to perform model selection to balance approximation and estimation error is important to improve sample efficiency while achieving low regret.

In Figure 4.5, we observe that IBES-32 does not have an decreasing regret as sample size increases in stochastic environments. We believe this phenomenon can be attributed primarily to two factors: (a) we report the top-1 regret so even minor variations in BE estimates might substantially change the top-1 policy, and (b) in stochastic environments, small models like IBES-32 may incur significant approximation errors, leading to a less pronounced reduction in generalization error despite increased sample sizes. In contrast, IBES models with larger hidden sizes do not demonstrate this behavior.

In summary, the experiment suggests the proposed BE method is more same efficient, due to the fact that the method predict the Bellman error, and the model selection procedure.

4.5.2 Comparison between FQE and IBES under different data coverage

In the second set of experiments, we aim to understand of how IBES and FQE perform when we vary important factors such as data coverage and sample size. We design two different datasets for the experiments: (a) well-covered data is

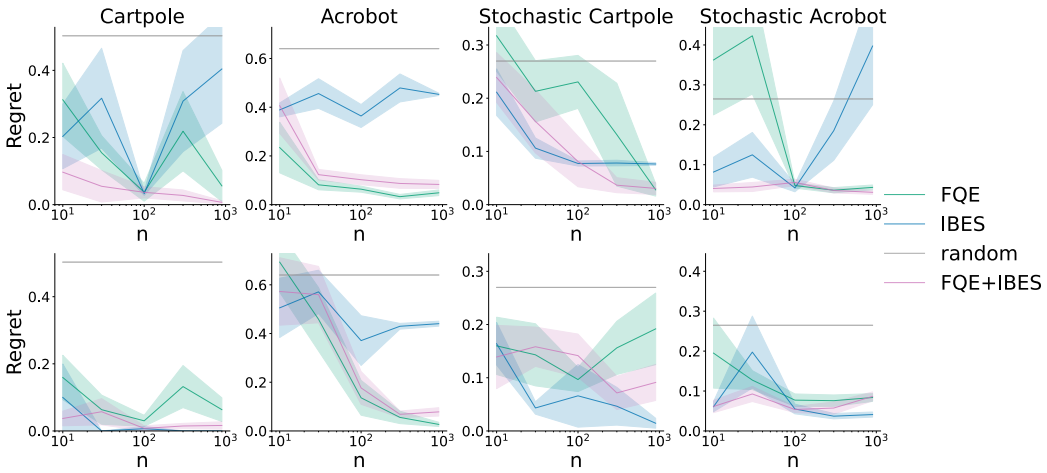


Figure 4.6: Comparison between IBES and FQE under different sample size and data distributions. The figure shows the normalized top-1 regret with varying sample size, averaged over 10 runs with one standard error.

generated such that all candidate policies are well-covered, and (b) well-covered data with optimal trajectories includes more diverse trajectories collected by an ε-greedy optimal policy (which used in the previous experiment).

Figure 4.6 shows the results for top-1 regret with varying numbers of episodes. We first focus on the asymptotic performance ($n \approx 10^3$) across different datasets. FQE performs very well with a small regret on well-covered and diverse data. IBES performs better with optimal trajectories, especially in Cartpole and Stochastic Acrobot. This result matches our theoretical result that IBES requires a stronger data coverage for an optimal policy. Moreover, IBES often performs better than FQE with a small sample size, suggesting that it could offer a slightly better sample efficiency than FQE.

Investigating the results deeper, we observed that when IBES does not perform well (Acrobot with optimal trajectories), it is often the case that one of the value functions has the smallest Bellman error but it is not far from optimal. When FQE does not perform well (Stochastic Cartpole with optimal trajectories), it is often the case that one of the candidate policies is highly overestimated. Therefore, we include a simple two-stage method that first uses FQE to select k_1 policies, then use IBES to find the top-k_2 policies amongst the k_1 selected policies. We set $k_1 = 10$ and $k_2 = 1$ in our experiment. The

idea is similar to the two-stage method proposed in Tang and Wiens (2021). We label the two-stage method as FQE+IBES in the figure. We can see that it performs consistently well, better than either method alone. We hypothesize the explanation is that FQE usually performs very well for top-k selection with a large k, even though top-1 regret might be bad. We then use IBES to select from a subset of reasonably good candidate policies, where the candidate with the smallest error is more likely to be optimal. Further investigation about combining multiple OPS methods might be a promising research direction.

4.5.3 Comparison between FQE and IBES on Atari

Finally, we conduct experiments on benchmark Atari datasets to show the hardness of OPS for offline RL, and compare IBES to FQE in a more practical setting. We use the offline data on Breakout and Seaquest from the DQN replay dataset[3], a commonly used benchmark in offline RL. We sample 1 million transitions from different learning stages to promote data coverage. Unlike our previous experiment, the data coverage might be poor due to the absent of explicit exploration to cover all candidate policies. We use 50% of the data to generate a set of candidate policy-value pairs by running CQL with different number of gradient steps and different regularization coefficients, as specified in Kumar et al. (2020). We use the other 50% data to perform OPS using FQE or IBES. We generate two candidate sets for each environment. The early-learning candidate contains policies with gradient steps in $\{50k, \cdots, 500k\}$ and final-learning candidate contains policies with gradient steps in $\{550k, \ldots, 1000k\}$. In this experiment, we use the CQL and FQE implementation from the d3rlpy library (Seno & Imai, 2022).

Figure 4.7 shows the top-1 regret. We can see that these none of the OPS methods can consistently outperform the random baseline, showing the hardness of OPS in a practical situation where we can not control the data coverage. For Breakout with early-learning candidate set, we found that IBES tends to pick the candidate value function with a small number of gradient steps. This is likely due to the fact that none of the candidate value functions

[3]https://research.google/resources/datasets/dqn-replay/

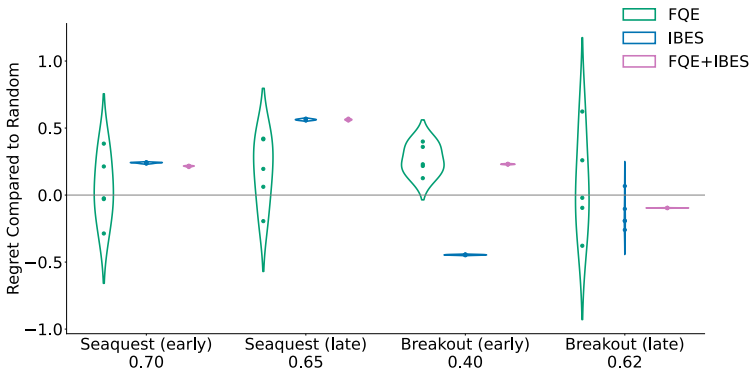


Figure 4.7: Regret improvement over the random baseline on the Atari dataset. We show the regret compared to the random baseline. A positive value means the method outperforms random selection, and a negative value means the method performs worse than random selection. We show the distribution of the regret improvement across 5 random seeds (each seed is a single point in the violin plots) and the regret of the random baseline under the x-axis labels. None of these methods can consistently outperform the random baseline, showing the hardness of OPS.

are close to optimal (that is, large ε_{sub}) and the value function with a small number of training steps has a small magnitude and hence a small estimated Bellman error. IBES performs better with the candidate set containing policies in the final learning stage, where it is more likely to contain a value function that is close to optimal. This shows the limitation of IBES, as discussed in the previous section. We can also see that FQE is more robust to the choice candidate set since it does not require one of the value function being optimal, as long as all candidate set are covered by the data. However, it is more sensitive to the data used to run FQE, and hence high variance across different random seeds.

4.6 Related work

In this section we provide a more comprehensive survey of prior work on model selection for RL. In the online setting, model selection has been studied extensively across contextual bandits (Foster et al., 2019) to RL (J. Lee et al., 2021). In the online setting, the goal is to select model classes while balancing exploration and exploitation to achieve low regret, which is very different from

the offline setting where no exploration is performed.

In the offline setting, besides using OPE and BE selection, other work on model selection in RL is in other settings: selecting models and selecting amongst OPE estimators. Hallak et al. (2013) consider model selection for model-based RL algorithms with batch data. They focus on selecting the most suitable model that generates the observed data, based on the maximum likelihood framework. Su, Srinath, et al. (2020) consider adaptive estimation ion for OPE when the candidate estimators can be ordered with monotonically increasing biases and decreasing confidence intervals.

In offline RL pipelines, we often split the offline data into training data for generating multiple candidate policies and validation data for selecting the best candidate policy. Nie et al., 2022 highlight the utility of performing multiple random data splits for OPS. They do not study the hardness or sample complexity of this procedure.

To the best of our knowledge, there is no previous work on understanding the fundamental limits for OPS in RL. There is one related work in the batch contextual bandit setting, studying the selection of a linear model (J. Lee et al., 2022). They provide a hardness result suggesting it is impossible to achieve an oracle inequality that balances the approximation error, the complexity of the function class, and data coverage. Our work considers the more general problem, selecting a policy from a set of policies, in the RL setting.

4.7 Conclusion

In this chapter, we made contributions towards understanding when OPS is feasible for RL. One of our main results—that the sample complexity of OPS is lower-bounded by the sample complexity of OPE—is perhaps expected. However, to our knowledge, this has never been formally shown. This result implies that without conditions to make OPE feasible, we cannot do policy selection efficiently. Our second contribution is the proposed IBES algorithm. We provide a sample complexity analysis, and empirically show that it is more sample efficient than existing BE-based methods for OPS.

We expect active research topics will be (1) to identify suitable conditions on the policies, environments and data, to make OPS sample efficient, (2) to design offline RL algorithms that have sound methods to select their own hyperparameters, and (3) to investigate how to combine multiple methods for a better OPS algorithm. In offline RL, we cannot select hyperparameters by testing in the real-world, and instead are limited to using the offline data. OPS is arguably one of the most critical steps towards bringing RL into the real-world, and there is much more to understand.

Chapter 5

Exploiting Exogenous Structures for Offline RL

In Chapter 3, we discussed several negative results of offline RL. These results motivate the need to look at specific classes of MDPs where offline RL can be sample efficient. In this chapter, we explore a restricted class of MDPs to obtain guarantees for offline RL. The key property, which we call Action Impact Regularity (AIR), is that actions primarily impact a part of the state (an endogenous component) and have limited impact on the remaining part of the state (an exogenous component). AIR is a strong assumption, but it nonetheless holds in a number of real-world domains including financial markets. We present an algorithm that exploits the AIR property, called FQI-AIR, and provide an analysis on the sample complexity of FQI-AIR. Finally, we demonstrate that the algorithm outperforms existing offline RL algorithms across different data collection policies in simulated and real-world environments where the regularity holds.

5.1 Introduction

We have discussed common conditions for sample efficient RL, such as a small concentration coefficient or data coverage for an optimal policy. Intuitively, there are settings where offline RL should be effective while these assumptions do not hold. Consider a trading agent in a stock market. A policy that merely observes stock prices and volumes without buying or selling any shares

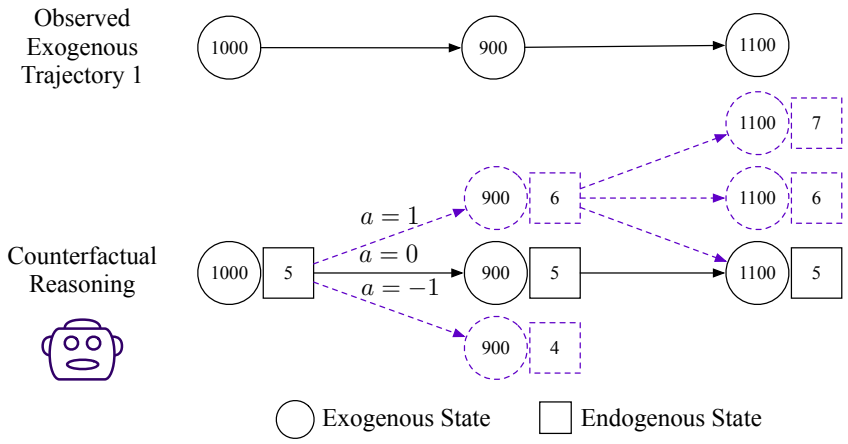


Figure 5.1: In the stock market example, the exogenous state corresponds to the stock price, the endogenous state corresponds to number of shares the agent has, and the action corresponds to the number of share to buy or sell at each time step. Given an observed exogenous trajectory, the agent can counterfactually reason about the outcomes of different actions and endogenous state.

provides useful information about the environment. For this collected dataset, an offline agent can counterfactually reason about the utility of many different actions as demonstrated in Figure 5.1, because its actions have limited impact on the prices and volumes. Such MDPs, which are called Exogenous MDPs, have states that separate into exogenous states (stock price), not impacted by actions, and endogenous states (number of shares owned by the agent). The structure in Exogenous MDPs has been used in online RL to learn more efficiently (Dietterich et al., 2018).

This exogenous structure, however, has yet to be formally investigated for offline RL, though it is likely already being exploited in industry. Exploiting this structure is natural in applied financial applications, because datasets allow for alternative trajectories to be simulated, as described in the above example. One (unpublished) system uses RL and trajectory simulation for the optimal order execution problem (Burhani et al., 2020); it seems likely that there are other such systems in use. What has yet to be done, however, is to understand the theoretical properties of such algorithms, as well as potential algorithmic improvements.

In this chapter, we first generalize the definition of exogenous MDPs, and

formalize the action impact regularity (AIR) property. We say an MDP has the AIR property—or is ε-AIR—if the actions have a limited impact on exogenous dynamics, with the level of impact determined by $\varepsilon \geq 0$. This generalizes the previous definition, which required strict separation, corresponding to $\varepsilon = 0$. We develop both theory and algorithms for this more general setting, assuming access only to an offline dataset, an approximate (learned) endogenous model and the reward function.[1]

We design an efficient algorithm, called FQI-AIR, to exploit the AIR property, that (1) does not require an estimate of the behavior policy or the data distribution, (2) has a straightforward approach to select hyperparameters using just the given data and (3) is much less sensitive to the quality of the data collection policy, if our assumptions hold. This algorithm is a simple extension of FQI, but is significantly more computationally efficient than the trajectory simulation approach mentioned above and allows us to leverage and extend the existing theory for FQI. We bound the suboptimality of the output policy from FQI-AIR, in terms of ε and other standard terms such as model errors and the inherent Bellman error. Importantly, in place of the concentration coefficient, we have a term that depends on the size of the endogenous state and number of actions; when the concentration coefficient is bounded and small, it is on the same order as this term.

We then conduct a comprehensive empirical study of FQI-AIR. We compare several algorithms in two simulated environments, across three different data collection policies, with varying offline dataset sizes, for $\varepsilon = 0$ (assumption perfectly satisfied) and a larger ε (assumption somewhat violated). FQI-AIR significantly outperforms the offline RL algorithms that do not leverage the AIR property—including FQI, MBS-QI, CQL and IQL; this outcome is expected, but nonetheless verifies that exploiting the AIR property, when appropriate, can have a big benefit. We show that these conclusions extend to two environments based on real-world datasets (for bitcoin trading and for

[1] We could assume an approximate instead of exact reward model. However, in most RL analysis, the error on the transition model is of a higher order than the error for reward models (for example, see A. Agarwal et al. (2020)). For simplicity, it is often assumed the reward model is known.

controlling battery usage in a hybrid car). An important detail here is how hyperparameters are chosen. FQI-AIR can exploit AIR for policy evaluation, to automatically select hyperparameters. For the other algorithms, we do not have such an approach, and instead report idealized performance by picking hyperparameters based on performance in the environment.

Finally, these results all used the true endogenous model for FQI-AIR. We chose to do so partly because the endogenous model is known for certain AIR-MDPs (e.g., trading, inventory management) and partly to focus the investigation on the role of ε rather than model error. However, for certain AIR-MDPs, we will not have access to the true endogenous model. In our final experiment, we investigated the impact of using a learned endogenous model in the hybrid car environment, and show that FQI-AIR remains effective.

All of this is only possible because we make a strong assumption about the environment. However, given the hardness results in offline RL, we should acknowledge that we likely need to restrict the class of MDPs. This work is a step towards understanding for what classes of MDPs offline RL is feasible. At the same time, though we consider a restricted setting, it is by no means a trivial setting. There are many real-world examples where this regularity holds (as we discuss later in this chapter). This is doubly true given that our generalization provides some flexibility in violating the assumption: the regularity only needs to hold approximately rather than exactly. The algorithms and theory developed here can benefit these real-world applications now, by providing an approach that is well-designed and well-behaved with strong theoretical guarantees for their specific problem setting.

5.2 Action impact regularity

Actions play an important role for the exponential lower bound constructions cited in the last section. These lower bounds use tree structures where different actions lead to different subtrees and hence different sequence of futures states and rewards. A class of MDPs that do not suffer from these lower bounds are those where actions do not have such strong impact on the future states and

rewards. In this section, we introduce the Action Impact Regularity (AIR) property, a property of the MDP which allows for more effective offline RL. The state is partitioned into an exogenous and endogenous component, and the property reflects that the agent's actions primarily impact the endogenous state with limited influence on the exogenous state. We first provide the formal definition and assumptions we leverage to design a practical offline RL algorithm and then discuss when these assumptions are likely to be satisfied.

5.2.1 Formal definition and assumptions

We use the standard state decomposition from Exogenous MDPs (Dietterich et al., 2018; McGregor et al., 2017). We assume the state space is $\mathcal{S} = \mathcal{S}^{\text{exo}} \times \mathcal{S}^{\text{end}}$ where $\mathcal{S}^{\text{exo}}$ is the exogenous variable and $\mathcal{S}^{\text{end}}$ is the endogenous variable. The transition dynamics are $P^{\text{exo}} : \mathcal{S}^{\text{exo}} \times \mathcal{A} \to \Delta(\mathcal{S}^{\text{exo}})$ and $P^{\text{end}} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S}^{\text{end}})$ for exogenous and endogenous variable respectively. The transition probability from a state $s_1 = (s_1^{\text{exo}}, s_1^{\text{end}})$ to another state $s_2 = (s_2^{\text{exo}}, s_2^{\text{end}})$ is $P(s_1, a, s_2) = P^{\text{exo}}(s_1^{\text{exo}}, a, s_2^{\text{exo}}) P^{\text{end}}(s_1, a, s_2^{\text{end}})$.

Definition 4 (The AIR Property)**.** An MDP is ε-AIR if $\mathcal{S} = \mathcal{S}^{\text{exo}} \times \mathcal{S}^{\text{end}}$, and for any actions $a, a,' \in \mathcal{A}$, the next exogenous variable distribution is similar if either action a or a' is taken. That is, for each state $s \in \mathcal{S}$,

$$D_{TV}\left(P^{\text{exo}}(s^{\text{exo}}, a), P^{\text{exo}}(s^{\text{exo}}, a')\right) \leq \varepsilon$$

where D_{TV} is the total variation distance between two probability distributions on $\mathcal{S}^{\text{exo}}$. For discrete spaces, the total variation distance is $D_{TV}(P, P') = \frac{1}{2}\|P - P'\|_1$ (ℓ_1 norm).

We define the AIR-MDP such that the property holds for all exogenous state-action pairs. If the property does not hold for one of the exogenous state-action pairs, then one can design an adversarial MDP that hides all difficulties in this single exogenous state-action pair and assuming the properties hold for all but one pair would be useless (Jiang, 2018).

Access to an (approximate) endogenous model is critical to exploit the AIR property, and is a fundamental component of our algorithm. To be precise, we make the following assumption in this chapter.

Assumption 1 (AIR with an Approximate Endogenous Model)**.** We assume that the MDP is ε_{air}-AIR and that we have the reward model $r : \mathcal{S} \times \mathcal{A} \to [0, r_{max}]$ and an approximate endogenous model $\hat{P}^{\text{end}} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S}^{\text{end}})$ such that $D_{TV}(P^{\text{end}}(s,a), \hat{P}^{\text{end}}(s,a)) \leq \varepsilon_p$ for any $(s,a) \in \mathcal{S} \times \mathcal{A}$.

As mentioned in the introduction, it is common to assume that only the transition dynamics are approximated. Moreover, similar to Definition 4, we need the error on the approximate model to hold uniformly. Finally, the above assumption implicitly assumes that the separation between exogenous and endogenous state is given to us. More generally, the separation could be identified or learned by the agent, as has been done for contingency-aware RL agents (Bellemare et al., 2012) and wireless networks (Dietterich et al., 2018). Because there are many settings where the separation is clear, we focus on this more clear case first where the separation is known.

5.2.2 When are these assumptions satisfied?

Many real-world problems can be formulated as ε-AIR MDPs. Further, for many of these environments, the separation between exogenous and endogenous state is clear, and we either know or can reasonably approximate the endogenous model. In this section, we go through several concrete examples.

We can first return to our stock trading example, from the introduction. The exogenous component is the market information (stock prices and volumes) and the endogenous component is the number of stock shares owned by the agent. The agent's actions influence their own number of shares, but as an individual trader, have limited impact on stock prices. Using a dataset of stock prices over time allows the agent to reason counterfactually about the impact of many possible trajectories of actions (buying/selling) on its shares (endogenous state) and profits (reward).

There are many settings where the agent has a limited impact on a part of the state. The optimal order execution problem is a task to sell M shares of a stock within H steps; the goal is to maximize the profit. The problem can be formulated as an MDP where the exogenous variable is the stock price

and endogenous variable is the number of shares left to sell. It is common to assume infinite market liquidity (Nevmyvaka et al., 2006) or that actions have a small impact on the stock price (Abernethy & Kale, 2013; Bertsimas & Lo, 1998); this corresponds to assuming the AIR property.

Another example is the secretary problem (Freeman, 1983), which a family of problems that can often be used to model real-world application (Babaioff et al., 2007; Goldstein et al., 2020). The goal for the agent is to hire the best secretary out of H, interviewed in random order. After each interview, they have to decide if they will hire that applicant, or wait to see a potentially better applicant in the future. The problem can be formulated as a 0-AIR MDP where the endogenous variable is a binary variable indicating whether we have chosen to stop or not.

Other examples include those where the agent only influences energy efficiency, such as in the hybrid vehicle problem (Lian et al., 2020; Shahamiri, 2008) and electric vehicle charging problem (Abdullah et al., 2021). In the former problem, the agent controls the vehicle to use either the gas engine or the electrical motor at each time step, with the goal to minimize gas consumption; its actions do not impact the driver's behavior. In the latter problem, the agent controls the charging schedule of an electric vehicle to minimize costs; its actions do not impact electricity cost.

In some settings we can even restrict the action set or policy set to make the MDP ε-AIR. For example, if we know that selling M shares hardly impacts the markets, we can restrict the action space to selling less than or equal to M shares. In the hybrid vehicle example, if the driver can see which mode is used, we can restrict the policy set to only switch actions periodically to minimize distractions for the driver.

In these problems with AIR, we often know the reward and transitions for the endogenous variables, or have a good approximation. For the optimal order execution problem, the reward is simply the selling price times the number of shares sold minus transaction costs, and the transition probability for P^{end} is the inventory level minus the number of shares sold. In other applications, we may be able to use domain knowledge to build an accurate model for the

endogenous dynamics. For the hybrid vehicle, we can use domain knowledge to calculate how much gas would be used for a given acceleration. Such information about the dynamics of the system can be simpler for engineers to specify, than (unknown) behavior of different drivers and environment conditions. Our theoretical results will include a term for the error in the endogenous model, but it is reasonable to assume that for many settings we can get that error to be relatively low, particularly in comparison to the error we might get if trying to model the exogenous state.

5.2.3 Connections to the literature on exogenous MDPs

AIR MDPs can be viewed as an extension of Exogenous MDPs. (1) We allow the action to have small impact on the environmental state, while the action has no impact on the exogenous state in Exogenous MDPs. (2) We do not assume the reward can be decomposed additively to an exogenous reward and an endogenous reward (Dietterich et al., 2018) nor factor into a sum over each exogenous state variable (Chitnis & Lozano-Pérez, 2020). For this previous definition of Exogenous MDPs, the focus was on identifying and removing the exogenous state/noise so that the learning problem could be solved more efficiently (Dietterich et al., 2018; Efroni et al., 2022; Efroni et al., 2021). Our focus is offline learning where we want to exploit the known structure to enable counterfactual reasoning and avoid data coverage issues.

Weakly-coupled MDPs (Meuleau et al., 1998), share similarities with exogenous MDPs. Weakly-coupled MDPs can be decomposed into independent sub-MDPs where the reward function and the transition dynamics of each sub-MDP are independent of the others. These sub-MDP are linked only via global resource constraints. In such models, it is typical to assume *additive utility independent*, meaning that the global reward is the sum of individual reward associated with each sub-MDP. Therefore, once we allocate the global resources to these sub-problems, we can then solve these sub-problems independently. In contrast, our AIR MDPs differ in two critical ways: (1) the endogenous component can depend on the exogenous component. (2) We do not assume the reward can be decomposed additively into exogenous and en-

dogenous components.

5.3 Offline policy learning for AIR MDPs

In this section, we discuss several offline algorithms that exploit the AIR property for policy learning. We then theoretically analyze an FQI-based algorithm, characterizing the performance of its outputted policy.

5.3.1 Algorithms for AIR MDPs

Two standard classes of algorithms in offline RL are model-based algorithms—that learn a model from the offline dataset and then use dynamic programming—and model-free algorithms like fitted Q-iteration (FQI). These two approaches can be tailored to our setting with AIR MDPs, as we described below. There is, however, an even more basic approach in our offline RL setting using trajectory simulation, that has previously been used (Burhani et al., 2020). We start by describing this simpler approach, and then the modified model-based and FQI approaches.

A natural approach is to reuse trajectories in the dataset to simulate alternative trajectories for an online RL algorithm. For each episode, a random trajectory is selected from the dataset. The online RL algorithm—such as an actor-critic method or a Q-learning agent—takes actions and deterministically transitions to the next exogenous state in the trajectory. The approximate endogenous and reward model are used to sample the next endogenous variable and reward. With such a trajectory simulator, we can run any online reinforcement learning algorithm to find a good policy for the simulator.

This approach, however, does not exploit the fact that the agent is actually learning offline. The online RL algorithm cannot simply query the model for any state and action, and needs a good exploration strategy to find a reasonable policy. There are fewer theoretical guarantees for such online RL algorithms, and arguably more open questions about their properties than DP-based algorithms and fitted value iteration algorithms.

A more explicit model-based approach is to learn the exogenous model from

data, to obtain a complete transition and reward model, and use any planning approach. The transition model for exogenous states can be constructed as if the action has no impact. With the model, we can use any query-efficient planning algorithm to find a good policy for the model. Because actions have only small impact in the true MDP, we can learn an accurate exogenous model even if we do not have full data coverage.

More precisely, recall the offline data is randomly generated by running π_b on M, that is, $D = \{(S_0^{(i)}, A_0^{(i)}, \ldots, S_{H-1}^{(i)}, A_{H-1}^{(i)})\}_{i=1}^N$ sampled according to the probability measure $\mathbb{P}_M^{\pi_b}$. The pertinent part is the transitions between exogenous variables, so we define $D_{\text{exo}} = \{(S_0^{(i)}, S_1^{(i),\text{exo}}, \ldots, S_{H-1}^{(i),\text{exo}})\}_{i=1}^N$. The model-based approach constructs an empirical MDP $M_D = (\mathcal{S}, \mathcal{A}, \hat{P}, r, H, \hat{\nu})$. For the tabular setting we have $\hat{\nu}(s) = \frac{1}{N}\sum_{i=1}^N \mathbb{I}(S_0^{(i)} = s)$, and

$$\hat{P}^{\text{exo}}(s_h^{\text{exo}}, a, s_{h+1}^{\text{exo}}) = \frac{\sum_{i=1}^N \mathbb{I}(S_h^{(i),\text{exo}} = s_h^{\text{exo}}, S_{h+1}^{(i),\text{exo}} = s_{h+1}^{\text{exo}})}{\sum_{i=1}^N \mathbb{I}(S_h^{(i),\text{exo}} = s_h^{\text{exo}})}$$

for all $a \in \mathcal{A}$. Exogenous variables not seen in the data are not reachable, and so can either be omitted from $\hat{P}^{\text{exo}}$ or set to self-loop. For large or continuous state spaces, we can learn $p(s_{h+1}^{\text{exo}}|s_h^{\text{exo}})$ using any conditional distribution learning algorithm, and set $\hat{P}^{\text{exo}}(s_h^{\text{exo}}, a, s_{h+1}^{\text{exo}}) = p(s_{h+1}^{\text{exo}}|s_h^{\text{exo}})$ for all $a \in \mathcal{A}$.

For large or continuous states spaces, however, learning such a model and planning can be impractical. Learning an accurate exogenous model might be difficult if the exogenous transition is complex or the exogenous state is high-dimensional. Further, it is not possible to sweep through all states during planning. Smarter approximate dynamic programming algorithms need to be used, but even these can be quite computationally costly.

A reasonable alternative is FQI, which approximates value iteration without the need to learn a model. Our FQI algorithm that exploits the AIR property is described in Algorithm 4, which we call FQI-AIR. The algorithm simulates all actions from a state, and assumes it transitions to the exogenous state observed in the dataset. The reward and endogenous state for each simulated action can be obtained using the reward model and approximate endogenous model. Even though the true MDP is not necessarily 0-AIR MDP, we will show in the analysis that as long as ε_{air} is small, the algorithm can

Algorithm 4 FQI-AIR

Input: dataset D, value function class $\mathcal{F}$, $\hat{P}^{\text{end}}$, r
Let $q_H = 0$, $D_{H-1} = \emptyset$, ..., $D_1 = \emptyset$
for $h = H-1, \ldots, 0$ **do**
For all $i \in \{1, \ldots, N\}$, all $s_h^{\text{end}} \in \mathcal{S}^{\text{end}}$, all $a \in \mathcal{A}$
Sample $s'^{\text{end}} \sim \hat{P}^{\text{end}}(s_h^{(i),\text{exo}}, s_h^{\text{end}}, a)$, compute target

$$t = r(s_h^{(i),\text{exo}}, s_h^{\text{end}}, a) + \max_{a' \in \mathcal{A}} q_{h+1}(s_{h+1}^{(i),\text{exo}}, s'^{\text{end}}, a')$$

Add (synthetic) pair $((s_h^{(i),\text{exo}}, s^{\text{end}}, a), t)$ to D_h
After generating D_h

$$q_h = \arg\min_{f \in \mathcal{F}} \sum_{(x,y) \in D_h} (f(x) - y)^2$$
$$\pi_h(s) = \arg\max_a q_h(s, a) \text{ for all } s \in \mathcal{S}_h$$

Output: $\pi = (\pi_0, \ldots, \pi_{H-1})$

return a nearly optimal policy in the true MDP. This algorithm, although simple, enjoys theoretical guarantees without making assumptions on the concentration coefficient, and can be much more computationally efficient than trajectory simulation methods.

Note that the computational cost scales with the size of $\mathcal{S}^{\text{end}}$ and $\mathcal{A}$. When $|\mathcal{S}^{\text{end}}|$ or $|\mathcal{A}|$ is large, we can modify FQI-AIR to no longer use full sweeps. Instead, we can randomly sample from the endogenous state space and action. We include a practical implementation of FQI-AIR in Algorithm 5. For each exogenous state in the dataset, we sample an endogenous state and an action, and query the approximate model to obtain a target for FQI update. As a result, the computation can be independent of the size of $\mathcal{S}^{\text{end}}$ and $\mathcal{A}$. However, for sample complexity, the performance loss of the algorithm would depend on the squared root of the size $|\mathcal{S}^{\text{end}}||A|$, as shown in the next section.

5.3.2 Theoretical analysis of FQI-AIR

First we need the following definitions. For a given MDP M, we define $J(\pi, M) := \mathbb{E}_M^\pi[R(\tau)]$ where $\tau = (S_0, A_0, \ldots, S_{H-1}, A_{H-1})$ is a random ele-

Algorithm 5 FQI-AIR for large state spaces

Input: dataset D, an approximate model $\hat{P}^{\text{end}}$, r, mini-batch size B, number of training iteration K, number of updates per iteration M
Initialize a Q function $q_\theta : \mathcal{S}^{\text{exo}} \times \mathcal{S}^{\text{end}} \times \mathcal{A} \times [H] \to \mathbb{R}$, parameterized by θ
$\bar{\theta} \leftarrow \theta$
for $k = 1, \ldots, K$ **do**
  for $m = 1, \ldots, M$ **do**
    Sample a mini-batch of transitions $\{(s_j^{\text{exo}}, s_j^{\text{end}}, a_j, h_j, s_j'^{\text{exo}}, s_j'^{\text{end}})\}_{j=1}^B$ from D
    For all j, sample an endogenous state $\tilde{s}_j^{\text{end}} \in \mathcal{S}^{\text{end}}$ and an action $\tilde{a}_j \in \mathcal{A}$ randomly, sample $\tilde{s}_j'^{\text{end}} \sim \hat{P}^{\text{end}}(s_j^{\text{exo}}, \tilde{s}_j^{\text{end}}, \tilde{a}_j)$, and compute target

$$t_j = r(s_j^{\text{exo}}, \tilde{s}_j^{\text{end}}, \tilde{a}_j) + \max_{a' \in \mathcal{A}} q_{\bar{\theta}}(s_j'^{\text{exo}}, \tilde{s}_j'^{\text{end}}, a', h_j + 1)$$

    Compute the mini-batch loss $L(\theta) = \sum_{j=1}^B (q_\theta(s_j^{\text{exo}}, \tilde{s}_j^{\text{end}}, \tilde{a}_j, h_j) - t_j)^2$
    Update θ to reduce $L(\theta)$
  $\bar{\theta} \leftarrow \theta$
Output: the greedy policy with respect to q_θ

ment in $(\mathcal{S} \times \mathcal{A})^H$, the expectation $\mathbb{E}_M^\pi$ is with respect to $\mathbb{P}_M^\pi$, and $R(\tau) = \sum_{h=0}^{H-1} r(S_h, A_h)$.

We also need the following assumption on the function approximation error. This is a common assumption to analyze approximate value iteration algorithms (Antos et al., 2008; Munos, 2007). Let $\tilde{\nu}_h(s_h^{\text{exo}}) = \mathbb{P}_M^{\pi_b}(S_h^{\text{exo}} = s_h^{\text{exo}})$ be the data distribution on $\mathcal{S}^{\text{exo}}$ at horizon h. Given a probability measure ν_h on $\mathcal{S}^{\text{exo}}$ and $p \in [1, \infty)$, define the weighted norm as

$$\|q\|_{p,\nu_h}^p = \sum_{s^{\text{exo}} \in S^{\text{exo}}} \sum_{s^{\text{end}} \in S^{\text{end}}} \sum_{a \in \mathcal{A}} \frac{\nu_h(s^{\text{exo}})}{|\mathcal{S}^{\text{end}}||\mathcal{A}|} |q(s^{\text{exo}}, s^{\text{end}}, a)|^p.$$

Assumption 2. Assume the function class $\mathcal{F}$ is finite and the inherent Bellman error is bounded by ε_{apx}, that is,

$$\varepsilon_{apx} = \max_{h \in [H]} \max_{f' \in \mathcal{F}} \min_{f \in \mathcal{F}} \|\mathcal{T} f' - f\|_{2,\tilde{\nu}_h}^2.$$

We assume the function class is finite for simplicity, which is common in many offline RL papers (Chen & Jiang, 2019). If the function class is not

finite but has a bounded complexity measure, we can derive similar results by replacing the size of the function class with the complexity measure. For example, Duan et al. (2021) analyze FQI with the Rademacher complexity. Since studying the complexity measure is not a critical point in this chapter, we decide to make the finite function class assumption.

Theorem 4 (Performance bound for FQI-AIR)**.** Under Assumption 1 and 2, let π^*_M be an optimal policy in M and π the output policy of FQI-AIR, then with probability at least $1-\zeta$

$$J(\pi, M) \geq J(\pi^*_M, M) - 2v_{max}H(\varepsilon_{air} + \varepsilon_p) - \\ (H+1)H\sqrt{|\mathcal{S}^{\text{end}}||\mathcal{A}|}\left(\sqrt{\frac{72{v_{max}}^2 \ln\left(H|\mathcal{F}||\mathcal{S}^{\text{end}}||\mathcal{A}|/\zeta\right)}{N}} + 2\varepsilon_{apx}\right).$$

The bound on performance loss has three components: (1) a sampling error term which decreases with more trajectories; (2) the AIR parameter; and (3) an approximation error term which depends on the function approximation used. The result implies that as long as we have a sufficient number of episodes, a good function approximation, and small ε_{air}, then the algorithm can find a nearly-optimal policy with high probability. For example, if $\varepsilon_{air}, \varepsilon_p$ and ε_{apx} are small enough, we only need $N = \tilde{O}(H^4 {v_{max}}^2 |\mathcal{S}^{\text{end}}||\mathcal{A}|/\delta)$ trajectories, which is polynomial in H, to obtain a δ-optimal policy.

The proof can be found in Appendix B.1. The key idea is to introduce a baseline MDP M_b that is 0-AIR, that approximates M which is actually ε_{air}-AIR. The baseline MDP $M_b = (\mathcal{S}, \mathcal{A}, \tilde{P}, r, H, \nu)$ has

$$\tilde{P}^{\text{exo}}(s^{\text{exo}}_h, a, s^{\text{exo}}_{h+1}) = \mathbb{P}^{\pi_b}_M(S^{\text{exo}}_h = s^{\text{exo}}_h, S^{\text{exo}}_{h+1} = s^{\text{exo}}_{h+1}) / \mathbb{P}^{\pi_b}_M(S^{\text{exo}}_h = s^{\text{exo}}_h)$$

and $\tilde{P}^{\text{end}}(s_h, a, s^{\text{end}}_h) = \hat{P}^{\text{end}}(s_h, a, s^{\text{end}}_h)$. The transition probability for exogenous state does not depend on the action a taken, so M_b is 0-AIR. We show that FQI returns a good policy in M_b, and that good policies in M_b are also good in the true MDP M.

We can contrast this bound to others in offline RL. For FQI results that assume the concentration coefficient is bounded and small, the error bound has a term that scales with $\sqrt{C}$, which is on the same order as the term $\sqrt{|\mathcal{S}^{\text{end}}||\mathcal{A}|}$

in our bound. We can get a similar bound by considering this restricted class of MDPs that are ε-AIR, without having to make any assumptions on the concentration coefficient. For settings where this assumption is appropriate—namely when the MDP is ε-AIR—this is a significant success, as we need not make these stringent conditions on data distributions.

5.4 Policy evaluation for AIR MDPs

We can also exploit the AIR property, and access to the approximate endogenous model and reward model, to evaluate the value of a given policy. Given a trajectories of exogenous states $(S_0^{(i)}, S_1^{(i),\text{exo}}, \ldots, S_{H-1}^{(i),\text{exo}})$, we can rollout a synthetic trajectory under π and $\hat{P}^{\text{end}}$: $\tau_D^{(i)} = (S_0^{(i)}, A_0^{(i)}, S_1^{(i)}, A_1^{(i)}, \ldots)$ where $A_t^{(i)} \sim \pi(S_t^{(i)})$, $S_{t+1}^{(i),\text{end}} \sim \hat{P}^{\text{end}}(S_t^{(i)}, A_t^{(i)})$ and $S_{t+1}^{(i)} = [S_{t+1}^{(i),\text{exo}}, S_{t+1}^{(i),\text{end}}]$. For $R(\tau_D^{(i)}) := \sum_{t=0}^{H-1} r(S_t^{(i)}, A_t^{(i)})$, the average return over the N synthetic trajectories $\hat{J}(\pi, M) = \frac{1}{N}\sum_{i=1}^{N} R(\tau_D^{(i)})$ is an estimator of $J(\pi, M)$. This method is simple, but very useful because we can do hyperparameter selection with only the offline dataset without introducing extra hyperparameters.

We can bound the policy evaluation error by Hoeffding's inequality. More sophisticated bounds for policy evaluation can be found in (P. Thomas et al., 2015a).

Theorem 5. Under Assumption 1, given a deterministic policy π, we have that with probability at least $1 - \zeta$

$$\left|\hat{J}(\pi, M) - J(\pi, M)\right| \leq v_{max}\left(H\varepsilon_{air} + H\varepsilon_p + \sqrt{\frac{\ln(2/\zeta)}{2N}}\right).$$

The results suggests that if we have a sufficient number of trajectories and small ε_{air} and ε_p, then $\hat{J}(\pi, M)$ is a good estimator for $J(\pi, M)$. Even though the estimator is biased and not consistent, we find it provides sufficient information for hyperparameter selection in our experiments.

Theorem 5 only holds for a given policy that is independent of the offline data. If we want to evaluate the output policy, which depends on the data, we need to apply an union bound for all deterministic policies. In that case, the sampling error term becomes $\tilde{O}(\sqrt{|\mathcal{S}|/N})$. To avoid the dependence on

the state space, an alternative is to split the data into two subsets: one subset is used to obtain an output policy and another subset is used to evaluate the output policy.

5.5 Simulation experiments

We evaluate the performance of the FQI-based algorithm in two simulated environments with the AIR property: an optimal order execution problem and an inventory management problem. FQI-AIR is notably better than these other approaches, and so we focus on it as the representative algorithm that exploits the AIR property.

The first goal of these experiments is to demonstrate that existing offline RL algorithms fail to learn a good policy for some natural data collection policies, while our proposed algorithm returns a near-optimal policy. To demonstrate this, we test three data collection policies: (1) a random policy which is designed to give a small concentration coefficient, (2) a learned near-optimal policy obtained using DQN with 1000 episodes of online interactions, which covers an optimal policy reasonably well, and (3) a constant policy which, in theory, has an infinite concentration coefficient due to missing state-action pairs and does not cover an optimal policy. The second goal is to validate the policy evaluation analysis with a varying number of trajectories N and ε_{air}.

5.5.1 Environments

We investigated the behavior of the algorithms on two simulated environments that mimic two real-world problems that satisfy the AIR property: optimal order execution and inventory management. In the **optimal order execution problem**, the task is to sell $M = 10$ shares within $H = 100$ steps. The stock prices $X_1, \ldots, X_h$ are generated by an ARMA$(2, 2)$ process and scaled to the interval $[0, 1]$. Specifically, the ARMA$(2, 2)$ process is

$$X_t = \tilde{X}_t/20 + 1/2 \qquad \text{where} \quad \tilde{X}_t = c + \varepsilon_t + \sum_{i=1}^{2} \varphi_i \tilde{X}_{t-i} + \sum_{i=1}^{2} \theta_i \varepsilon_{t-i},$$

$\varphi_1 \sim \mathcal{U}(-0.9, 0.0)$, $\varphi_2 \sim \mathcal{U}(0.0, 0.9)$ and $\theta_i \sim \mathcal{U}(-0.5, 0.5)$ for $i = 1, 2$. The scaling parameters are chosen so that the process is stable and the price is in the interval $[0, 1]$. The endogenous variable P_h is the number of shares left to sell. To construct state, we use the most recent $K = 3$ prices with the most recent endogenous variable, that is, $S_h = (X_{h-2}, X_{h-1}, X_h, P_h)$. The action space is $\mathcal{A} = [5]$. The reward function is the stock price X_h multiplied by the number of selling shares $\min\{A_h, P_h\}$.

We consider both a setting with $\varepsilon_{air} = 0$ and $\varepsilon_{air} > 0$, as well as different data generating policies. When the number of selling shares is greater than 0, the stock price drops by 10% with probability ε_{air}. For $\varepsilon_{air} = 0$, this means selling shares has no impact on the stock price. When $\varepsilon_{air} > 0$, it does, allowing us to test how robust FQI-AIR is to some violation of the AIR property. The random policy used in the environment chooses 0 with probability 75% and choose $1, \ldots, 5$ with probability 5%. The constant policy always chooses action 0.

We design an **inventory management problem** based on existing literature (Kunnumkal & Topaloglu, 2008; Van Roy et al., 1997). The task is to control the inventory of a product over $H = 100$ stages. At each stage, we observe the inventory level X_t (endogenous) and the previous demand D_{t-1} (exogenous) and choose to place a order $A_t \in [10]$. The inventory level evolves as: $X_{t+1} = (X_t + A_t - D_t)^+$. The reward function is $cA_t + h(X_t + A_t - D_t)^+ - b(-X_t - A_t + D_t)^+$ where c is the order cost, h is the holding cost, and b is the cost for lost sale. We use $c = 0.1$, $h = 0.25$ and $b = 1.0$ in the experiment. To make sure the reward is bounded, we clip the reward at a large negative number -100. The endogenous variable is the inventory level, which can be as large as 1000, so we restrict FQI-AIR to sweep only for a subset of the endogenous space, that is, $[15] \subset \mathcal{S}^{\text{end}}$.

As before, we consider both $\varepsilon_{air} = 0$ and $\varepsilon_{air} > 0$, which in this case impacts the demand. The demand $D_t = (\tilde{D}_t)^+$ where $\tilde{D}_t$ follows a normal distribution with mean μ and $\sigma = \mu/3$ and $(d)^+ := \max\{d, 0\}$. In the beginning of each episode, μ is sampled from a uniform distribution in the interval $[3, 9]$. When the order is greater than 0, the mean of the demand distribution decreases or

increases by 10% with probability $\varepsilon_{air}/2$ respectively.

Again, we consider three different data generating policies. The random policy used in the environment is a policy which chooses a value $\tilde{A}_t \in [D_{t-1} - 3, D_{t-1} + 3]$ uniformly and then choose the action $A_t = \max\{\min\{\tilde{A}_t, 10\}, 0\}$. The constant policy always chooses action $A_t = \min\{D_{t-1}, 10\}$. The near-optimal policy is obtained using DQN with online interaction, for both environments.

There is an important nuance for the inventory problem. In this problem, the endogenous transition and reward depends on the next exogenous variable. Fortunately, we can generalize the definition of exogenous MDPs such that the endogenous transition is $P^{\text{end}} : \mathcal{S} \times \mathcal{A} \times \mathcal{S}^{\text{exo}} \to \Delta(\mathcal{S}^{\text{end}})$ and the reward is $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S}^{\text{exo}} \to [0, r_{max}]$. We assume we have an approximate endogenous model $\hat{P}^{\text{end}} : \mathcal{S} \times \mathcal{A} \times \mathcal{S}^{\text{exo}} \to \Delta(\mathcal{S}^{\text{end}})$ such that $D_{TV}(P^{\text{end}}(s, a, s^{\text{exo}}), \hat{P}^{\text{end}}(s, a, s^{\text{exo}})) \leq \varepsilon_p$ for any $(s, a, s^{\text{exo}}) \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}^{\text{exo}}$. With these changes, the algorithms and the theoretical analysis naturally extend to the new definition of exogenous MDPs.

5.5.2 Algorithm details

We compare FQI-AIR to FQI, MBS-QI (Y. Liu et al., 2020), CQL (Kumar et al., 2020), IQL (Kostrikov et al., 2022). As we discussed in the previous sections, FQI is expected to work well when the concentration coefficient is small. MBS-QI is expected to perform well when the data covers an optimal policy. CQL and IQL are strong baselines which have been shown to be effective empirically for discrete-action environments such as Atari games.

We had several choices to make for the baseline algorithms. MBS-QI requires density estimation for the data distribution μ. For the optimal order execution problem, we use state discretization and empirical counts to estimate the data distribution as used in the original paper. For the inventory problem, the state space is already discrete so there is no need for discretization. We show the results with the best threshold b from the set $\{0.002, 0.001, 0.0001, 0.00005\}$. Note that it is possible that there is no data for some states (or state discretization) visited by the output policy, and for

these states, all action values are all zero. To break ties, we allow MBS-QI to choose an action uniformly at random. The original MBS-QI algorithm does not work with negative rewards. The reward in the inventory problem is in the range $[-100, 0]$, so we modify the pessimistic value for MBS-QI for the problem:

$$\tilde{q}_h(s,a) := \begin{cases} q_h(s,a), & \text{if } \hat{\mu}(s,a) \geq b \\ -10000, & \text{if } \hat{\mu}(s,a) < b \end{cases}$$

instead of $\tilde{q}_h(s,a) := \mathbb{I}\{\hat{\mu}(s,a) \geq b\} q_h(s,a)$ where $\hat{\mu}$ is the estimated data distribution.

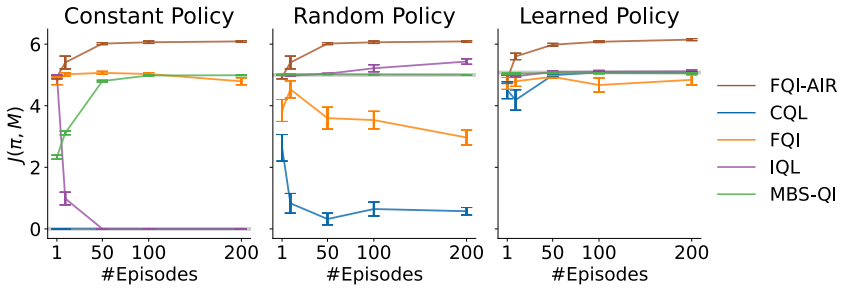
For CQL, we add the CQL($\mathcal{H}$) loss with a weighting α when updating the action values. We show the results with the best α from the set $\{0.1, 0.5, 1.0, 5.0\}$ as suggested in the original paper. For IQL, we show the results with the best τ from the set $\{0.7, 0.8, 0.9\}$ and β from the set $\{10.0, 3.0, 1.0\}$.

We use the same value function approximation for all algorithms in our experiments: two-layers neural networks with hidden size 128. The neural networks are optimized by Adam (Kingma & Ba, 2014) or RMSprop with learning rate in the set $\{0.001, 0.0003, 0.0001\}$. All algorithms are trained for 100 iterations. We also tried training the comparator algorithms for longer, but it did not improve their performance.

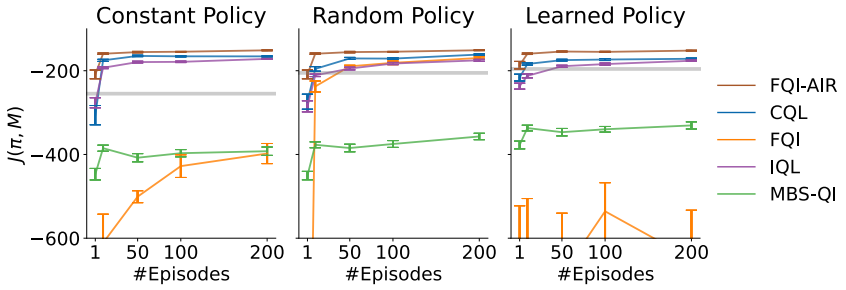
The hyperparameters for FQI-AIR are selected based on $\hat{J}(\pi, M)$, which only depends on the dataset. The hyperparameters for comparator algorithms are selected based on $J(\pi, M)$—which should be a large advantage—estimated by running the policy π on the true environment M with 100 rollouts.

5.5.3 Policy performance when $\varepsilon_{air} = 0$

Figure 5.2 shows the performance of our algorithm and the comparator algorithms with a different number of trajectories $N = \{1, 5, 10, 25, 50, 100, 200\}$ and $\varepsilon_{air} = 0$. Our algorithm outperforms other algorithms for all data collection policies. This result is not too surprising, as FQI-AIR is the only algorithm to exploit this important regularity in the environment; but nonetheless it shows how useful it is to exploit this AIR property when it is possible.



(a) Optimal order execution problem



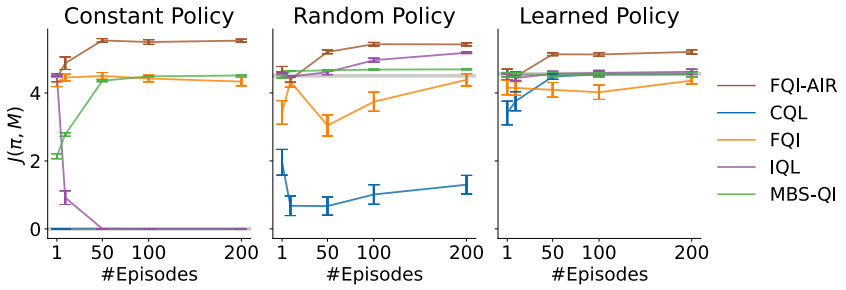
(b) Inventory management problem

Figure 5.2: Comparison between algorithms in the optimal order execution problem and inventory management problem, for $\varepsilon_{air} = 0$. The gray lines show the performance of the data collection policies. Results are averaged over 30 runs with error bars for one standard error.

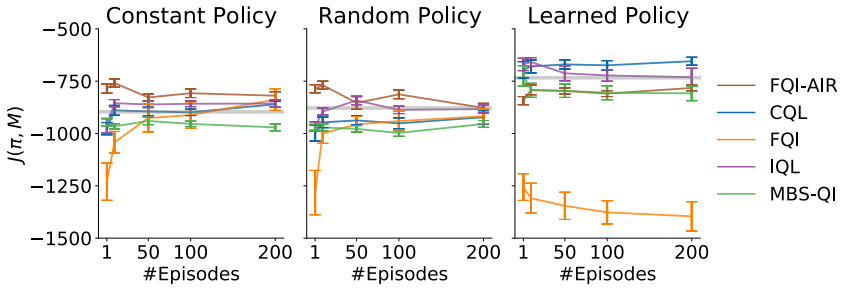
We can first look more closely at the optimal order execution results. MBS performs slightly better than FQI, however, we found it performs better because the tie-breaking is done with a uniform random policy especially under the constant policy dataset.[2] CQL and IQL fails when the data collection policy is far from optimal (constant policy) and perform reasonably with a learned policy. FQI-AIR exploits the AIR property, and so is robust to different data collection policies. The results show that exploiting the AIR property is critical for the robust performance.

We see similar patterns for the inventory management problem. FQI-AIR outperforms the other algorithms for all data collection policies. CQL and IQL perform well in this environment. MBS outperforms FQI under the learned policy, but FQI outperforms MBS under the random policy. The results match the expectation that FQI performs well with an uniform data and MBS-QI performs well with an expert data.

[2]A uniform policy in this environment can achieve a performance $J(\pi, M) \approx 5$.



(a) Optimal order execution problem



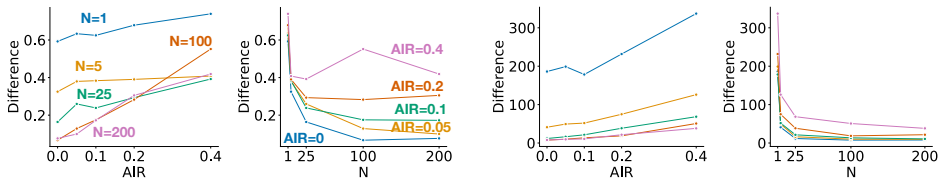
(b) Inventory management problem

Figure 5.3: Comparison between algorithms in two simulation problems with $\varepsilon_{air} = 0.8$. The gray lines show the performance of the data collection policies. The results are averaged over 30 runs with error bars representing one standard error.

5.5.4 Policy performance with a large value of ε_{air}

Now we consider the impact of using these algorithms when $\varepsilon_{air} > 0$. We should expect FQI-AIR to be most impacted, as the other algorithms do not exploit the AIR property. We vary ε_{air} from 0.1 to 0.8 and find that the results are similar to those with small ε_{air}. FQI-AIR still significantly outperforms other offline methods.

Figure 5.3 shows the result with $\varepsilon_{air} = 0.8$ where the performance of all algorithms drop significantly. In theory, FQI-AIR can have a large performance loss with large ε_{air}, however, FQI-AIR still consistently outperforms other baselines in our experiments, except for the inventory management problem with the learned policy. This is because the divergence between the true exogenous transition and the synthetic exogenous transition in FQI-AIR does not occur at every time step even when ε_{air} is large. For example, in the optimal order execution problem, the divergence can only happen when we sell



(a) Optimal order execution problem (b) Inventory management problem

Figure 5.4: The 90th percentile, over 90 runs, of the difference $|\hat{J}(\pi, M) - J(\pi, M)|$, with varying N and ε_{air}. The difference should be lower for a larger N and smaller ε_{air}.

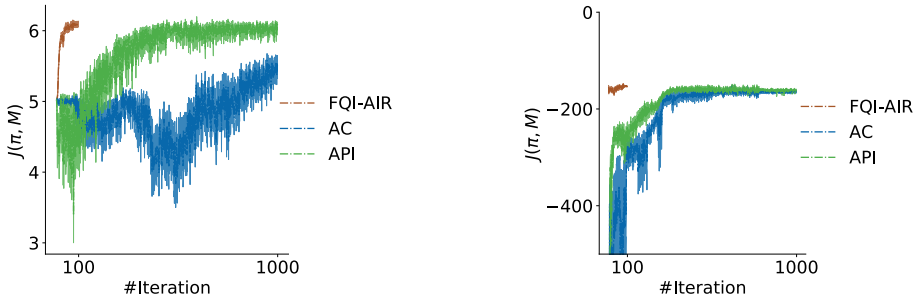
a positive number of shares. The theoretical result is the worst-case analysis where the divergence can occur at every time step and we suffer r_{max} loss every time the divergence occurs. Therefore, the experiment results suggest that these practical problems considered in the chapter are not the worst case and FQI-AIR can perform well even with large ε_{air}.

5.5.5 Results for policy evaluation

To validate the policy evaluation analysis, we investigate the difference $|\hat{J}(\pi, M) - J(\pi, M)|$ with $\varepsilon_{air} \in \{0, 0.05, 0.1, 0.2, 0.4\}$ and $N = \{1, 5, 25, 100, 200\}$ where π is the output policy of FQI-AIR. We show the 90th percentile of the difference for each combination of ε_{air} and N over 90 data points (30 runs under each data collection policy) in Figure 5.4. The 90th percentiles scale approximately linearly with ε_{air} and inversely proportional to N. The results suggest that the dependence on ε_{air} is linear and the policy evaluation error goes to zero at a sublinear rate, which reflects the bound of Theorem 5.

5.5.6 Comparing FQI-AIR to online RL with trajectory simulation

We tested the other algorithms described in Section 5.3.1. We compare to two online RL algorithms, approximate policy iteration (API) and actor critic (AC) with ε-greedy exploration, with the trajectory simulator discussed in Section 5.3.1. We show the learning curves with random data collection policy and $N = 100$. Each iteration contains a sweep over the entire dataset. The



(a) Optimal order execution problem (b) Inventory management problem

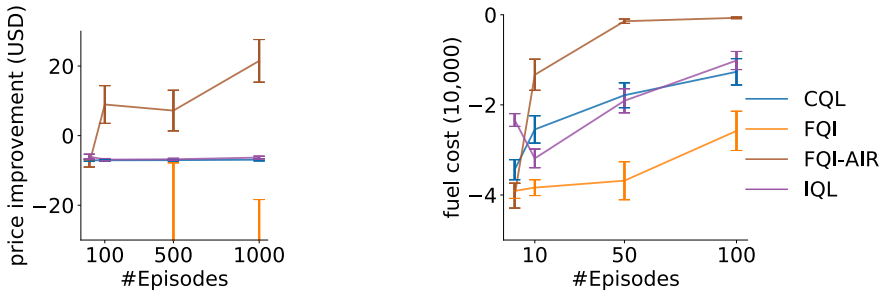
Figure 5.5: Comparison to simulation-based algorithms. Results are averaged over 30 runs with the shaded region representing plus and minus one standard error.

results show that FQI-AIR converges to a nearly optimal solution within a few iterations while the online RL algorithms require much more iterations to find a good policy. AC in the optimal order execution problem does not converge to a stable performance. The final performance of API and AC are also worse than the performance of FQI-AIR. These results show that online RL algorithms could be used for AIR MDPs. However, they are less direct and efficient, and they could find a slightly different solution with a finite amount of data and computation.

5.6 Real-world experiments

To demonstrate the practicality of the proposed algorithm, we evaluate the proposed algorithm for two real-world experiments: (1) *Bitcoin*: an optimal order execution for the bitcoin market, and (2) *Prius*: a hybrid car control problem. For the Bitcoin experiment, we use historical prices of bitcoin.[3] The problem is to sell one bitcoin within 60 steps where each step corresponds to 10 minutes in real world. On each step, the agent chooses to sell some numbers of bitcoins in $\{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$. Each episode corresponds to 10 hours, with a start state chosen from a random time step in the data (consisting of 300 days). The exogenous state contains the most recent 60 closing prices, and

[3]The bitcoin data is downloaded from the kaggle competition https://www.kaggle.com/c/g-research-crypto-forecasting/data.



(a) Bitcoin (b) Prius

Figure 5.6: Performance on real-world datasets. For (a), the numbers represent the total selling price minus the average price. For (b), the numbers represent the fuel cost with a penalty for not maintaining a desired batter level. Results are averaged over 30 runs, shown with one standard error.

the endogenous state contains the number of shares left to be sold. We collect an offline dataset by running a trained policy by DQN for N episodes, and report performance of the output policy for the testing period (about 41 days).

For the Prius experiment, we use the hybrid car environment from Lian et al. (2020).[4] The agent can switch between using fuel or battery, with the goal to minimize fuel consumption while maintaining a desired battery level. The exogenous state is the driving patterns and the endogenous state contains the state of charge and the previous action. We collect the offline dataset by running a learned policy with 10 different driving patterns, and test on 12 driving patterns. To better mimic the real-world, where we would not have a random policy or constant policy, we use the learned policy from DQN as the data collection policy. Further, now that the state space is larger, we run FQI-AIR where we randomly sample endogenous states and actions, rather than sweeping through all endogenous states and actions.

FQI-AIR performs significantly better than CQL, IQL and FQI, as shown in Figure 5.6. MBS-QI does not scale to high-dimensional continuous state spaces, and so is excluded. These results highlight that FQI-AIR can scale to high-dimensional continuous state space and large endogenous state space.

[4]We used their code at https://github.com/lryz0612/DRL-Energy-Management/blob/master/Prius_model_new.py.

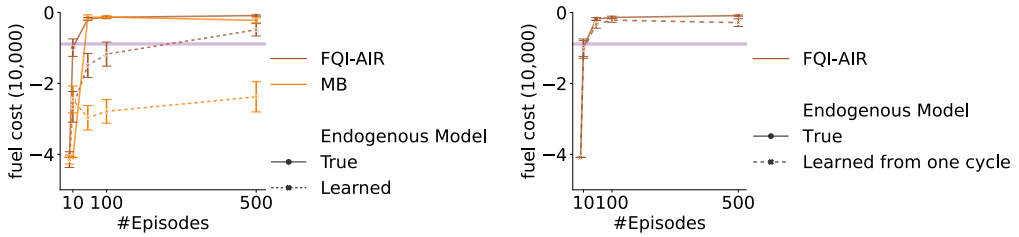
5.6.1 Learning an endogenous model in AIR-MDPs

In the previous experiments, we assume we are given the endogenous models. In this section, we investigate the impact of using an approximate endogenous model learned from offline data. We perform this experiment in the hybrid car environment, which reflects a setting where the endogenous dynamics might in fact not be known and would be useful to estimate from data. We use a neural network to approximate the endogenous state and reward model, and run FQI-AIR with the learned endogenous model.

Let us first reason about when it might be feasible to learn a reasonably accurate endogenous model. In the worse case, learning an accurate endogenous and reward model would require data coverage for the entire state space. However, in many practical scenarios, the endogenous model can be easy to learn and does not require full data coverage. For example, in the optimal order execution problem, the endogenous dynamics does not depend on the exogenous variables, as a result, we only need coverage for the endogenous state. In the Prius environment, collecting data from just one driving cycle is sufficient to learn a good endogenous model, as we will demonstrate in these experiments.

We first collect a dataset from the hybrid car environment by running a random policy in one of the driving cycles and a deterministic policy for the other driving cycles. This data generation approach mimics a scenario we might see in practice. In the factory, we might have a test system for which it is acceptable to try many different actions (using gas or the battery), and so get a more varied dataset for learning the endogenous model. We would only get this data from one limited course (one driving cycle). The rest of the data would be collected in the wild, where the deployed solution should not be exploring many actions and should largely be deterministic.

We also test two model-based baselines (MB): (1) The first baseline has full knowledge of the reward and endogenous models, and learns the exogenous model from offline data without exploiting the AIR property. The algorithm is similar to Algorithm 4 but s^{exo}_{h+1} is generated from the learned exogenous model.



(a) Comparison to model-based baselines

(b) Learning the endogenous model with data from one cycle

Figure 5.7: Experiment on the Prius dataset. We include IQL with 500 episodes as a baseline (purple line). Results are averaged over 30 runs, shown with one standard error.

(2) The second baseline does not have knowledge of the reward, endogenous models and exogenous models. It learns a full model to from a state-action pair to the next state. For these model-based baselines, the model is parameterized by a two-layer neural network and learned by minimizing the ℓ_2 distance between predicted states and next states recorded in the data. The transitions in these environments are deterministic, so it is appropriate to learn an expectation model.

In Figure 5.7 (a), we perform an ablation study to compare FQI-AIR and MB with the true endogenous model or a learned endogenous model. The result shows that (1) MB with the true endogenous model performs slightly worse than FQI-AIR with a small data size. (2) FQI-AIR with a learned endogenous model perform worse than FQI-AIR, however, it outperforms IQL and MB without the true endogenous model. (3) MB with a learned endogenous model performs worse than FQI-AIR with a learned endogenous model. This suggests that it is useful to separate the exogenous state and endogenous state especially when we need to learn an endogenous model.

Next, we test FQI-AIR when learning the endogenous model only from a more limited dataset: a dataset based solely on one cycle. We collect a dataset from the hybrid car environment by running a random policy in one of the driving cycles for 500 episodes. This reflects a practical scenario that we can just run our vehicle in a closed area and still are able to obtain a good endogenous model for running FQI-AIR. Figure 5.7 (b) shows that FQI-AIR

with the learned endogenous model performs well and is close to FQI-AIR with true endogenous model.

5.7 Conclusion

Our goal in this chapter is to study an MDP property that (1) is realistic for several important real-world problems and (2) makes offline RL feasible. We introduced an MDP property, which we call Action Impact Regularity (AIR). We developed an algorithm for MDPs satisfying AIR, that (1) has strong theoretical guarantees on the supoptimality, without making assumptions about concentration coefficients or data coverage, (2) provides a simple way to select hyperparameters offline, without introducing extra hyperparameters and (3) is simple to implement and computationally efficient. We showed empirically that the proposed algorithm significantly outperforms existing offline RL algorithms, across two simulated environments and two real-world environments.

Chapter 6

Dealing with Nonstationarity for Offline RL

In this chapter, we consider the off-policy policy evaluation (OPE) problem for contextual bandits and finite horizon RL in the non-stationary setting. Reusing old data is critical for policy evaluation, but existing estimators that reuse old data introduce large bias such that we can not obtain a valid confidence interval. Inspired from a related field called survey sampling, we introduce a variant of the doubly robust (DR) estimator, called the regression-assisted DR estimator, that can incorporate the past data without introducing a large bias. The estimator unifies several existing OPE methods and improves on them with the use of auxiliary information and a regression approach. We prove that the new estimator is asymptotically unbiased, and provide a consistent variance estimator to a construct a large sample confidence interval. Finally, we empirically show that the new estimator improves estimation for the current and future policy values, and provides a tight and valid interval estimation in several non-stationary recommendation environments.

6.1 Introduction

OPE is the problem of estimating the expected return of a target policy from a dataset collected by a different behavior policy. OPE has been used successfully for many real-world systems, such as recommendation systems (Li et al., 2011) and digital marketing (P. S. Thomas et al., 2017), to select a good policy

to be deployed in the real world. A variety of estimators have been proposed, particularly based on importance sampling (IS) (Hammersley & Handscomb, 1964) and modifications to reduce variance, such as self-normalization (Swaminathan & Joachims, 2015b), direct methods that use reward models and variance reduction techniques like the doubly robust (DR) estimator (Dudik et al., 2011; Jiang & Li, 2016; P. Thomas & Brunskill, 2016). Often high-confidence estimation is key, with the goal to estimate confidence intervals around these value estimates that maintain coverage without being too loose (Kuzborskij et al., 2021; Swaminathan & Joachims, 2015a; P. Thomas et al., 2015a, 2015b).

Much less work has been done, however, for the non-stationary setting where the reward and transition dynamics change over time. Extending these approaches to the non-stationary setting is key as most real-world systems change with time, or appear to due to partial observability. In this setting, we face a critical bias-variance tradeoff: using past data introduces bias, but not using past data introduces variance. Jagerman et al. (2019) introduced the sliding-window IS and exponential-decay IS estimator, that gradually reduces the impact of older data to control the bias-variance tradeoff. There is some other work predicting future OPE values for a target policy in a non-stationary environment, by using time-series forecasting (Chandak et al., 2020; P. S. Thomas et al., 2017); the goal there, however, is to forecast future policy values using past value estimates, rather than to estimate the current value.

Much of the other work tackling non-stationary problems has been for policy optimization. There is a relatively large body of work on non-stationary bandits in the on-policy setting (e.g., see J. Y. Yu and Mannor, 2009). More pertinent to this work is a recent approach in the off-policy setting (Hong et al., 2021). Their focus, however, is on the use of change point detection and hidden Markov models for policy optimization in the online phase. As a result, these ideas do not directly apply to non-stationary OPE.

In this chapter, we consider the piecewise stationary setting with known change points. This setting can be viewed as a special case of the multi-task settings, where we have offline data gathered from different intervals or "tasks". Lazaric et al. (2008) introduce a algorithm designed to transfer samples from

different tasks to the source task, which can be potentially used in the piecewise stationary setting. Nevertheless, it's important to note that this method might still result in considerable bias.

In this chapter, we propose a new approach for non-stationary OPE by exploiting ideas from a related field called *survey sampling* (Cochran, 1977), where handling non-stationary data has been a bigger focus. We propose a variant of the DR estimator, called the regression-assisted DR estimator, for non-stationary environments. We exploit two ideas: (1) using auxiliary variables from the past data to build a proxy value and incorporate the proxy value in the estimator without introducing bias, and (2) a regression approach on top of the proxy value to reduce variance further. Using the regression approach introduces some bias, however, we prove that the estimator is asymptotically unbiased and provide a consistent variance estimator to construct a large sample confidence interval (CI). Moreover, we show that this regression-assisted estimator unifies several existing OPE methods, including the weighted IS estimator. We empirically show that in several recommendation problems, formalized as contextual bandits, that the new estimator improves the estimation and provides a tighter and valid CI empirically compared to the sliding-window estimators. We then extend the idea to finite horizon RL, and highlight similar improvements.

6.2 Problem setup

In this section, we describe our main problem setting: OPE in the non-stationary setting. To convey the core insights of this chapter precisely, we first focus on contextual bandits.

Notation. We start by describing the standard stationary setting for OPE in the contextual bandit setting. Let $\mathcal{S}$ be a set of contexts, $\mathcal{A}$ be a set of actions, and $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ be the reward function. The goal is to evaluate the value of a target policy π, that is, estimate $J(\pi) = \mathbb{E}_{S\sim P, A\sim\pi(\cdot|S)}[r(S, A)]$, using an offline (off-policy) dataset. The dataset is created through the interaction of

a behavior policy with the environment: (1) the environment draws a context s_i from $P \in \Delta(\mathcal{S})$ and (2) the behavior policy draws an action a_i from $\pi_b(\cdot|s_i)$ and observes $r_i = r(s_i, a_i)$. This process repeats n times, giving dataset $D = \{(s_i, \mathbf{x}_i, a_i, r_i)\}_{i=1}^n$. We assume that we also observe the context feature $\mathbf{x}_s \in \mathbb{R}^d$ for each context s in the dataset.

Non-stationary OPE. Dealing with arbitrary nonstationarity may not be possible. Fortunately, many real-world environments have structures than can be exploited. We consider a piecewise stationary setting with known change points, where the reward function changes across intervals but remains stationary within each interval. For example, an environment can be stationary within each day or each week or for a number of interactions. We assume the set of contexts and the set of actions do not change over time.

Let r_k denote the reward function for the k-th interval and

$$D_k = \{(s_i, a_i, r_k(s_i, a_i))\}_{i=1}^{n_k}$$

denote the data of size n_k collected over the k-th interval. The goal is to estimate

$$J_k(\pi) = \sum_{s \in \mathcal{S}, a \in \mathcal{A}} P(s)\pi(a|s)r_k(s, a)$$

given previous datasets $D_1, \ldots, D_{k-1}$ and a newly sampled D_k. The problem mirrors the real world where we have plenty of past data $D_1, \ldots, D_{k-1}$ but only a small amount of new data D_k to estimate the current value $J_k(\pi)$. We consider a stationary context distribution to present this chapter succinctly, however, our methods described in this chapter are applicable to the settings where the context distribution is also changing.

It is often necessary in high-stakes applications to provide confidence intervals. Let $\mathcal{D} = (D_t)_{t=1}^k$ denote the set of all data collected across different intervals. Given $\mathcal{D}$ and a desired level of failure probability $\alpha \in (0, 1)$, it would be ideal to estimate a high-confidence lower bound CI^- and a high-confidence upper bound CI^+ such that

$$\Pr\left(\mathrm{CI}^-(\mathcal{D}, \alpha) \leq J_k(\pi) \leq \mathrm{CI}^+(\mathcal{D}, \alpha)\right) = 1 - \alpha$$

where the probability is under the randomness of D_k and conditional on all old data $D_1, \ldots, D_{k-1}$.

6.3 Estimators for stationary OPE and survey sampling

In this section, we review existing estimators for stationary OPE and describe how OPE can be written using the survey sampling formulation. We use this survey sampling formulation to introduce the proposed estimators in the next section.

6.3.1 Estimators for stationary OPE

A foundational strategy to estimate $J(\pi)$ in stationary OPE is to use importance sampling. The IS estimator is given by

$$\hat{J}_{\mathrm{IS}}(\pi) = \frac{1}{n}\sum_{i=1}^{n} \frac{\pi(a_i|s_i)}{\pi_b(a_i|s_i)} r(s_i, a_i).$$

This IS estimator can have high variance since the importance ratio can be very large. The weighted IS (WIS) estimator (Sutton & Barto, 2018), also known as the self-normalized estimator (Swaminathan & Joachims, 2015b), normalizes the importance weights and is more commonly used. The WIS estimator is given by

$$\hat{J}_{\mathrm{WIS}}(\pi) = \sum_{i=1}^{n} \frac{\pi(a_i|s_i)/\pi_b(a_i|s_i)}{\sum_{j=1}^{n} \pi(a_j|s_j)/\pi_b(a_j|s_j)} r(s_i, a_i).$$

Besides these IS-based estimators, another common estimator is the direct method (DM). We learn a reward prediction model $\hat{r}$ and use

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

The doubly robust (DR) estimator (Dudik et al., 2011) combines the DR and the IS estimator,

$$\hat{J}_{\mathrm{DR}}(\pi) = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{\pi(a_i|s_i)}{\pi_b(a_i|s_i)}(r(s_i, a_i) - \hat{r}(s_i, a_i)) + \sum_{a\in\mathcal{A}} \pi(a|s_i)\hat{r}(s_i, a)\right].$$

There are other OPE estimators such as estimators with clipping (Bottou et al., 2013) or shrinkage (Su, Dimakopoulou, et al., 2020). Dudik et al. (2012) studied the setting where the policies are non-stationary (history-dependent) but the environment is stationary, which is different from our setting. Chandak et al. (2021) focus on estimating the reward distribution and do not discuss how to efficiently leverage past data under non-stationarity.

6.3.2 OPE as survey sampling

Survey sampling can be dated back to Hansen and Hurwitz, 1943; Horvitz and Thompson, 1952, where they consider the problem of selecting a sample of units from a finite population to estimate unknown population parameters. Formally, let $\mathcal{U} = \{1, \ldots, N\}$ be the population of interest, y_i be the study variable and $\mathbf{x}_i$ be the auxiliary variable for the unit $i \in \mathcal{U}$. A subset of the population, called a sample, is selected according to a sampling design. We observe the study variable for units in the sample, and the goal is to estimate the population total of the study variables $t_y = \sum_{i \in \mathcal{U}} y_i$.

To formalize OPE under survey sampling, let the population be $\mathcal{U} = \mathcal{S} \times \mathcal{A}$ and the study variable be $y_{s,a} = P(s)\pi(a|s)r(s,a)$. The population total of y is the value of the policy: $t_y = \sum_{(s,a) \in \mathcal{U}} y_{s,a} = J(\pi)$. The weighting $P(s)\pi(a|s)$ goes into the study variable since the goal is to estimate the total of study variable without weighting. Even though we have $P(s)$ in the study variable, the term often cancels out in the estimator.

This formulation has some subtle differences from the standard OPE formulation. First, it assumes that $\mathcal{S} \times \mathcal{A}$ is finite, since $\mathcal{U}$ is finite in survey sampling. Second, the study variable is fixed, that is, the reward function is deterministic. These limitations can be overcome by assuming that the finite population is generated as a random sample from an infinite superpopulation; this superpopulation model is discussed in the appendix. For the main body, we assume a finite population with fixed study variables.

Of particular interest for nonstationarity is the *model-assisted* approach for survey sampling (Särndal et al., 1992). The key idea is to use the auxiliary variable $x_{s,a}$ to form a proxy value $\hat{y}_{s,a}$ such that $\hat{y}_{s,a}$ is close to the study

variable $y_{s,a}$. A simple example is that the auxiliary variable $x_{s,a}$ might be the value of $y_{s,a}$ at a past time and we can use proxy value $\hat{y}_{s,a} = x_{s,a}$. A general form for a model-assisted estimator, assuming the population total of the proxy value is known, is the difference estimator (Cassel et al., 1976): $\sum_{(s,a)\in\mathcal{U}} \hat{y}_{s,a} + \sum_{(s,a)\in D} \frac{y_{s,a}-\hat{y}_{s,a}}{np_{s,a}}$ where $p_{s,a}$ is the probability of selecting the pair (s, a). This estimator is unbiased and can be much lower variance, if the proxy value is close to the study variable. This strategy is like adding a control variate, but specific to survey sampling since the source of stochasticity is different than the typical Monte Carlo setting.

6.4 Regression-assisted estimators under non-stationarity

In the section, we propose an estimator for non-stationary environments. There are two popular strategies that consider the bias-variance tradeoff when reusing the past data in non-stationary environments: sliding window IS and exponential decay IS (Jagerman et al., 2019). The sliding window IS estimator directly uses the IS estimator for the data in the most recent B intervals. Though not proposed in the original work, it is natural to extend this idea to other estimators. For example, for the direct method, we can build a reward model from the data in the most recent B intervals and evaluate the policy with the reward prediction.

The window size B controls the bias-variance tradeoff. If $B = 0$ then we only use the most recent data D_k: the estimator does not introduce bias by using past data but suffers high variance due to having a small sample size. If we use a large B, the estimator might introduce large bias but might have lower variance due to a larger sample size. Sliding window estimators require carefully choosing B to balance the bias from using past data and the variance from not using past data. The balance usually depend on how fast the environment is changing, which is usually unknown. Moreover, even with a small value of B, the bias of the sliding window estimator can be so large that the confidence interval is invalid, as we will show in the experiment section.

Therefore, the main question that we aim to address is:

How can we reuse the past data for non-stationary OPE without introducing large bias?

One natural way to leverage the past data would be to use the DR estimator with a reward prediction learned from the past data, as described in the following section. However, naively using the past data to construct a reward prediction may not be the best approach in the non-stationary setting. This raises a followup question: *How can we obtain a good reward prediction to both reduce the error of estimation and also obtain tight CIs?* To address this challenge we draw inspiration from the survey sampling literature, and propose the regression-assisted DR estimator, that helps reduce variance further and provides tighter CIs.

6.4.1 The difference and DR estimator

We can leverage the idea of the difference estimator in survey sampling, for our non-stationary OPE setting. We can use the past data $D_{k-B}, \ldots, D_{k-1}$ to build a reward prediction $\hat{r}_k$ as a function of the context feature and the action: $\hat{r}_k(s,a) = m(\mathbf{x}_s, a; \theta)$ for some function m parameterized by θ. The reward prediction can be used as the proxy value in the estimator. The resulting difference estimator, for interval k, is

$$\hat{t}_{\text{Diff},k} = \sum_{(s,a)\in\mathcal{U}} P(s)\pi(a|s)\hat{r}_k(s,a) + \frac{1}{n}\sum_{(s,a)\in D_k} \frac{\pi(a|s)}{\pi_b(a|s)}(r(s,a) - \hat{r}_k(s,a)). \quad (6.1)$$

The elegance of this approach is that we can leverage past data by incorporating it into the proxy value in the difference estimator, without introducing any bias.

While the estimator is unbiased, the variance depends on the quality of the reward prediction $\hat{r}_k$ (P. Thomas & Brunskill, 2016). The environment is non-stationary, so past data has to be used carefully to get a good estimate, and in some cases, the estimate may be poor. In the next section, we discuss how to obtain a better prediction by fitting a regression on top of the reward prediction.

A careful reader would have noticed one other nuance with the above difference estimator: it requires the population total of the proxy value $\hat{y}_{s,a} = P(s)\pi(a|s)\hat{r}_k(s,a)$, which is the first term in Eq (6.1). In some cases, this information may be known and it should be leveraged to get a better estimator for OPE. In other cases, we will need to estimate it. In the standard contextual bandit setting, given a sample D_k, we often assume that we know the auxiliary variable $\mathbf{x}_{s,a}$ for all units in the set $\{(s,a) : s \in D_k, a \in \mathcal{A}\}$. If we estimate the population total from D_k with the information about the auxiliary variables, the estimator becomes

$$\hat{t}_{\mathrm{DR},k} = \frac{1}{n}\sum_{s\in D_k}\sum_{a\in\mathcal{A}}\pi(a|s)\hat{r}_k(s,a) + \frac{1}{n}\sum_{(s,a)\in D_k}\frac{\pi(a|s)}{\pi_b(a|s)}(r_k(s,a) - \hat{r}_k(s,a)). \tag{6.2}$$

This estimator reduces to the DR estimator. Therefore, the DR estimator is the difference estimator when the population total of the proxy value is estimated by sample D_k.

However, there are other options to estimate the population total, that do not result in the standard DR estimator. Of particular relevance here is that we can use past data D' to estimate this population total:

$$\frac{1}{|D'|}\sum_{s\in D'}\sum_{a\in\mathcal{A}}\pi(a|s)\hat{r}_k(s,a).$$

This term does not rely on rewards in the past data—which might not be correct due to nonstationarity—and only requires access to the auxiliary variables $\mathbf{x}_s$ in these datasets. If we assume only the rewards are non-stationary, rather than the context distribution, making these old datasets a perfectly viable option to estimate this population total. In survey sampling, this is usually motivated by assuming that there might be another survey that contains the auxiliary variables (S. Yang & Kim, 2020).

6.4.2 The regression-assisted DR estimator

We consider a model on top of the reward prediction from the past data to mitigate variance further. Let $\phi_k(s,a)^\top = (1, \hat{r}_k(s,a))$ be the augmented feature vector with the reward prediction and define $\mathbf{z}_{s,a} = P(s)\pi(a|s)\phi_k(s,a)$.

Note that $\mathbf{z}_{s,a}$ is a function of the auxiliary variable $\mathbf{x}_s$. We consider a (heteroscedastic) linear regression model such that the study variables $y_{s,a} := P(s)\pi(a|s)r_k(s,a)$ are realized values of the random variables $Y_{s,a}$ with $\mathbb{E}_\xi[Y_{s,a}] = \mathbf{z}_{s,a}^\top \beta$ and $\mathrm{V}_\xi(Y_{s,a}) = \sigma_{s,a}^2 = P(s)\pi(a|s)\sigma^2$ where the expectation and variance are with respect to the model ξ, and β, σ are the model coefficients. These are the assumptions underlying the regression estimator, rather than assumptions about the real world. Further, even though we consider a linear regression on the feature vector ϕ for the regression-assisted DR estimator, the reward prediction itself can be non-linear (e.g., a neural network).

The weighted least squares estimate of β is

$$\beta_k = \arg\min_\beta \sum_{(s,a)\in\mathcal{U}} \frac{1}{\sigma_{s,a}^2} \left(\mathbf{z}_{s,a}^\top \beta - y_{s,a}\right)^2.$$

Suppose the relevant matrix is invertible, β_k can be estimated using sample data D_k:

$$\hat{\beta}_k = \left(\sum_{(s,a)\in D_k} \frac{\pi(a|s)}{\pi_b(a|s)} \phi_k(s,a)\phi_k(s,a)^\top \right)^{-1} \left(\sum_{(s,a)\in D_k} \frac{\pi(a|s)}{\pi_b(a|s)} \phi_k(s,a) r_k(s,a) \right). \tag{6.3}$$

If we know the population total of $\mathbf{z}_{s,a}^\top \hat{\beta}_k$, then the regression-assisted DR (Reg) estimator is

$$\begin{aligned} \hat{t}_{\mathrm{Reg},k} = & \sum_{(s,a)\in\mathcal{U}} P(s)\pi(a|s)\phi_k(s,a)^\top \hat{\beta}_k \\ & + \frac{1}{n} \sum_{(s,a)\in D_k} \frac{\pi(a|s)}{\pi_b(a|s)} \left(r_k(s,a) - \phi_k(s,a)^\top \hat{\beta}_k \right). \end{aligned} \tag{6.4}$$

More generally, we can use the same data or the past data to estimate the population total, as described above.

This $\hat{\beta}_k$ consists only of the weight on the past reward model and the bias unit. This may not seem like a particularly useful addition, but because it is estimated using D_k, it allows us to correct the past reward prediction.

Further, the regression-assisted DR estimator actually provides a natural way to combine existing estimators in the OPE literature, depending on the

choice of the feature vector ϕ and the coefficients β. To see this, we first show how WIS can be seen as an instance of this estimator.[1]

We provide the theoretical result in the stationary setting where ϕ is fixed, so we can drop the subscript k for simplicity. For the non-stationary setting, the inference for $\hat{t}_{Reg,k}$ is conducted *conditional on* the past data $D_1, \ldots, D_{k-1}$, so ϕ is again fixed and all results extends to the non-stationary setting. The proofs can be found in Appendix C.2.

Theorem 6 (WIS as a special case of the regression-assisted estimator)**.** Suppose we use a linear regression model with univariate feature $\phi(s,a) = 1$. Then the regression-assisted DR estimator with estimated coefficient $\hat{\beta}$ from Eq (6.3) has the same form as the WIS estimator:

$$\hat{t}_{\text{Reg}} = \sum_{(s,a)\in D} \frac{\pi(a|s)/\pi_b(a|s)}{\sum_{(s',a')\in S} \pi(a'|s')/\pi_b(a'|s')} r(s,a). \tag{6.5}$$

The result provides a novel perspective for the WIS estimator: it can be viewed as fitting a regression to predict the reward with a constant feature. As a result, the only difference between the regression-assisted DR estimator and the WIS estimator is the choice of feature vector for reward prediction. If there are other features that might be useful for predicting the reward, we can include it with the regression approach and potentially improve the WIS estimator.

In Table 6.1, we show that we can recover other estimators based on different choices for the coefficients $\beta = (\beta_1, \beta_2)^\top$ with the feature vector $\phi(s,a)^\top = (1, \hat{r}(s,a))$. If $\beta_1 = 0, \beta_2 = 0$, we recover the IS estimator. If $\beta_1 = 0, \beta_2 = 1$, we recover the difference estimator or the DR estimator. If $\beta_2 = 0$ and β_1 is learned from data, we recover the WIS estimator.

There are other approaches to estimate the coefficients from data. The more robust DR estimator (Farajtabar et al., 2018) minimizes the estimated variance $\hat{\text{V}}(\hat{t}_{\text{Reg}})$ with respect to the coefficient to achieve the lowest asymptotic variance among all coefficient β under some mild conditions. Kallus and

[1] Mahmood et al., 2014 have a similar observation that the solution $\hat{\beta}$ is the WIS estimator if $\phi(s,a) = 1$ for all (s,a). They also extend the estimate with linear features ϕ and use $\phi(s,a)\hat{\beta}$ directly, which is more related to the model-based approach. In our work, the model prediction is used as the proxy value so the resulting estimators are different.

Table 6.1: A unifying view of existing estimators.

	IS	DR	WIS	Reg
β_1	0	0	$\hat{\beta}_1$	$\hat{\beta}_1$
β_2	0	1	0	$\hat{\beta}_2$

Uehara, 2019 further consider an expanded model class on top of the reward prediction and minimize the estimated variance among both the expanded model class and the reward prediction model class. However, it often unclear how large the sample size needs to be such that the estimator with the lowest asymptotic variance indeed has a lower variance against other estimators in practice. On the other hand, there is a considerable literature in survey sampling on improving estimation for the total and variance estimator when the sample size is small or the feature vector is high dimensional. For example, Breidt and Opsomer, 2000; K. S. McConville et al., 2017 propose different models as an alternative to the linear regression model. These regression models can be potentially more useful for feature selection or to find a model that fits the population well.

6.5 Theoretical analysis

In the regression approach, if the coefficients are estimated from the same data D_k, the estimator becomes biased. For example, the DR estimator is unbiased since the coefficients are fixed, and the WIS estimator is biased since one of the coefficients is estimated. In this section, we show that even if we run the regression on the same data we use to build the estimator, the regression-assisted DR estimator still enjoys asymptotic properties.

To prove these theoretical properties, there are a number of results from the survey sampling literature that we build on. For completeness, we provide a brief overview of survey sampling in Appendix C.1, and the proof of these properties under survey sampling notation in Appendix C.2.

Theorem 7 (Properties of the estimator)**.** Let $\mathrm{AV}(\cdot)$ denote the asymptotic variance in term of the first order, that is $\mathrm{V}(\cdot) = \mathrm{AV}(\cdot) + o(n^{-1})$, we have (1)

$\hat{t}_{\text{Reg}}$ is asymptotically unbiased with a bias of order $O(n^{-1})$, and (2)

$$\text{AV}(\hat{t}_{\text{Reg}}) = \frac{1}{n}\left(\sum_{(s,a)\in U} P(s)\frac{\pi(a|s)^2}{\pi_b(a|s)}(r(s,a) - \phi(s,a)^\top\beta)^2 - t_e^2\right)$$

where $t_e = \sum_{\mathcal{U}} P(s)\pi(a|s)(r(s,a) - \phi(s,a)^\top\beta)$.

Variance estimation for the regression-assisted DR estimator. The exact form of the variance of the regression-assisted DR estimator is often difficult to obtain, so we use the approximate variance from Theorem 7. Replacing the unknown β by the sample-based estimate $\hat{\beta}$, we have a variance estimator

$$\hat{\text{V}}(\hat{t}_{\text{Reg}}) = \frac{1}{n(n-1)}\left[\sum_{(s,a)\in D}\left(\frac{\pi(a|s)}{\pi_b(a|s)}(r(s,a) - \phi(s,a)^\top\hat{\beta})\right)^2 - n\hat{t}_e^2\right]$$

where $\hat{t}_e = \sum_D \frac{\pi(a|s)}{n\pi_b(a|s)}(r(s,a) - \phi(s,a)^\top\hat{\beta})$. Särndal et al., 1989 propose the weighted residual technique which can potentially result in better interval estimation.

Finally, we show the variance estimator is consistent and the regression-assisted estimator is asymptotically normal.

Theorem 8. The variance estimator $\hat{\text{V}}(\hat{t}_{\text{Reg}})$ is consistent, and $\frac{\hat{t}_{\text{Reg}} - t_y}{\sqrt{\hat{\text{V}}(\hat{t}_{\text{Reg}})}} \xrightarrow{D} \mathcal{N}(0,1)$.

Based on Theorem 8, we can construct a large sample CI.

Corollary 4. Let $\hat{\sigma} = \sqrt{\hat{\text{V}}(\hat{t}_{\text{Reg}})}$ and z_α denote the $100(1-\alpha)$ percentile of the standard normal distribution, then

$$\Pr\left(\hat{t}_{\text{Reg}} - z_{\alpha/2}\hat{\sigma} \le J(\pi) \le \hat{t}_{\text{Reg}} + z_{\alpha/2}\hat{\sigma}\right) \to 1 - \alpha.$$

6.6 Simulation and real-world experiments

In this section, we demonstrate the effectiveness of the regression-assisted DR estimators in a semi-synthetic and a real-world recommendation environment. We compare the proposed estimator to existing estimators, including IS, WIS, DM and Diff (which is DR without estimating the population total). We also

include the IS, WIS, DM with the sliding window (SW) approach of window size B. When $B = 0$, SW-IS and SW-WIS is the standard IS and WIS. For Diff, Reg, we use the past data $D_{k-B}, \ldots, D_{k-1}$ to learn a reward prediction.

For the semi-synthetic dataset, we follow the experimental design from Dudik et al., 2011. We use the supervised-to-bandit conversion to construct a partially labeled dataset from the YouTube dataset in the LibSVM repository. We construct a non-stationary environment by generating a sequence of reward functions based on the environment design in Chandak et al., 2020. For each true positive context-action pair in the original classification dataset, the reward follows a sine wave with some noise over time. We use PCA to reduce the dimension of the context features to 32. The target policy is obtained by training a classifier on a small subset of the original classification dataset.

We adapt the Movielens25m dataset (Harper & Konstan, 2015) for the real-world experiment. To construct a non-stationary environment, we divide the rating data chronologically. Each interval contains the rating data for 60 days and we use the rating data for $K = 24$ intervals ending November 21, 2019. We consider only active users who gave at least one rating for at least half of the K intervals, resulting in a total number of around 2000 users. During each interval, we compute the rating matrix $r(u, g)$ for each user and genre by averaging the user u's rating for all rated movies in the genre g. As a result, we have a sequence of rating functions which represent users' average rating for each genre over time. The user features are built by matrix factorization on the average rating data with hidden size 32, and the target policy is obtained by training a classifier on a small subset of the average rating data.

For the OPE objective, we consider an uniform weighting $P(s) = 1/|\mathcal{S}|$ for all users s. We also let $n_k = \alpha|\mathcal{S}|$ for all k and $\alpha \in \{0.1, 1.0\}$. For each interval $k = 0, \ldots, K$, we sample data D_k using a random policy. For estimators that require a reward prediction, we build the reward prediction by linear regression on historical data for each action separately, which is the same approach used in Dudik et al., 2011. More experiment details can be found in Appendix C.3, and Algorithm 6 describes the experimental procedure.

In non-stationary OPE, the aim is to estimate $J_k(\pi)$ with data from D_1

Algorithm 6 Non-stationary OPE experimental procedure

Input: dataset $(D_1, \ldots, D_k)$, target policy π with true policy value $J_1(\pi), \ldots, J_k(\pi)$, behavior policy π_b , prediction subroutine $Pred(X, Y, x_{test})$ using linear regression with basis function ψ
Hyperparameter: window size B
for $k = 1, \ldots, K$ **do**
  # Estimate $J_k(\pi)$ using regression-assisted DR estimator
  Compute $\hat{t}_{Reg,k}$ from $D_1, \ldots, D_k$
  # Predict $J_{k+1}(\pi)$ using linear regression
  $\hat{t}_{Pred,k+1} = Pred(X = [\psi(1), \ldots, \psi(k)], Y = [\hat{t}_{Reg,1}, \ldots, \hat{t}_{Reg,k}], x_{test} = \psi(k+1))$
Compute the true value $J_1(\pi), \ldots, J_k(\pi)$
Output:
RMSE for $J_k = \sqrt{\frac{1}{K}\sum_{k=1}^{K}(\hat{t}_{Reg,k} - J_k(\pi))^2}$
RMSE for $J_{k+1} = \sqrt{\frac{1}{K-d}\sum_{k=d+1}^{K}(\hat{t}_{Pred,k} - J_k(\pi))^2}$

to D_k. All of the estimators discussed in this chapter, however, can be extended to predict the future values using the ideas from Chandak et al. (2020) and P. S. Thomas et al. (2017). Suppose we have the OPE estimators for each interval up to interval k, that is, $\hat{J}_1(\pi), \ldots, \hat{J}_k(\pi)$, we can fit these data to a forecasting model to predict the future value $J_{k+1}(\pi), \ldots, J_{k+\delta}(\pi)$. We therefore test both settings: estimating $J_k(\pi)$ and predicting $J_{k+1}(\pi)$. For the experiments predicting $J_{k+1}(\pi)$, we adapt the method proposed in Chandak et al., 2020 and predict the future values by fitting a regression. That is, $\hat{J}_{k+\delta}(\pi) = \psi(k+\delta)^\top \hat{\mathbf{w}}_k$ where $\hat{\mathbf{w}}_k$ is the OLS estimator for the regression problem with feature map $\psi(t) = (cos(2\pi t n))_{n=0}^{d-1}$ and target $\hat{J}_t(\pi)$ for $t = 1, \ldots, k$, where we set $d = 5$ in the experiment. For clarity, we provide the pseudocode for our experiment procedure in Algorithm 6.

Sensitivity to window size and sample size. We vary the window size B and sample size n, and report the sensitivity plot in Figure 6.1. The error is averaged over K intervals, that is, RMSE $= \sqrt{\frac{1}{K}\sum_{k=1}^{K}(\hat{J}_k(\pi) - J_k(\pi))^2}$. We can see that the sliding window (SW) estimators, including SW-IS, SW-WIS and SW-DM, are sensitive to the window size, while Diff and Reg are robust to the window size. Reg outperforms IS and WIS and simply using $B = 1$

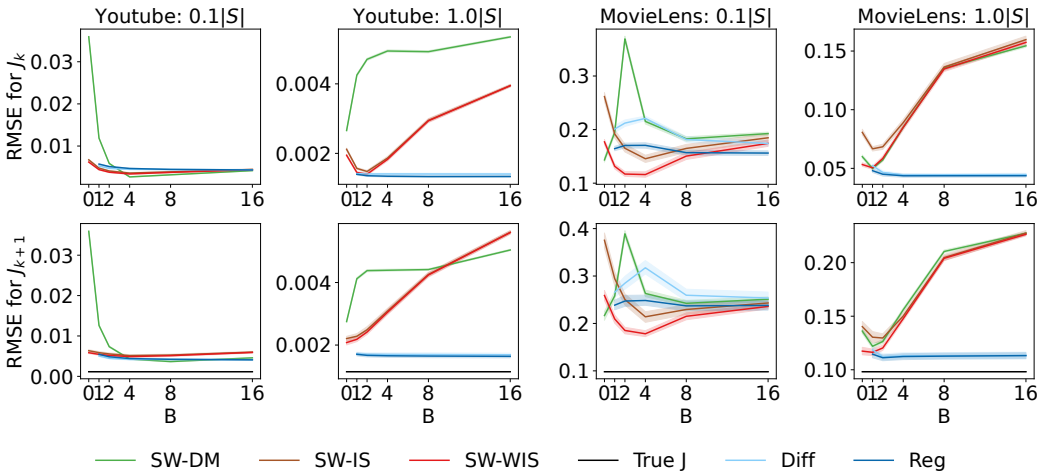


Figure 6.1: Sensitivity curves. **Top row: estimating $J_k(\pi)$. Bottom row: predicting $J_{k+1}(\pi)$.** "True J" is the baseline if we use the true values to predict the future values. The number are averaged over 30 runs with one standard error. Across runs, the target policy and the sequence of reward functions are fixed, but the sampled data is random.

reduces the error by a large margin. Reg also has a lower error compared to Diff, especially with small window size and sample size. This suggests that Reg is more robust to a bad reward prediction from the past data, which implies it is more robust to the speed of the nonstationarity.

We report the error for predicting the future value $J_{k+1}(\pi)$ in Figure 6.1. Reg has the lowest error for predicting the future values except in MovieLens with small sample size. We also find that even SW-DM and SW-IS have low error for estimating the current value J_k for some hyperparameters, they still have high error for predicting the future value J_{k+1}. We hypothesize that approximately unbiased estimators generally have better future prediction even though they might have high variance. It is possible that the forecasting model cancels out the noise in approximately unbiased estimators and results in better future value prediction.

Empirical validation of the interval estimation. We use $\hat{J}_k(\pi) \pm 1.96\sqrt{\hat{\mathbb{V}}(\hat{J}_k(\pi))}$ as the approximate 95% CI. We report the empirical coverage of the CI using the estimated variance in Figure 6.2. The empirical coverage is defined as the number of rounds such that the CI contains the true value

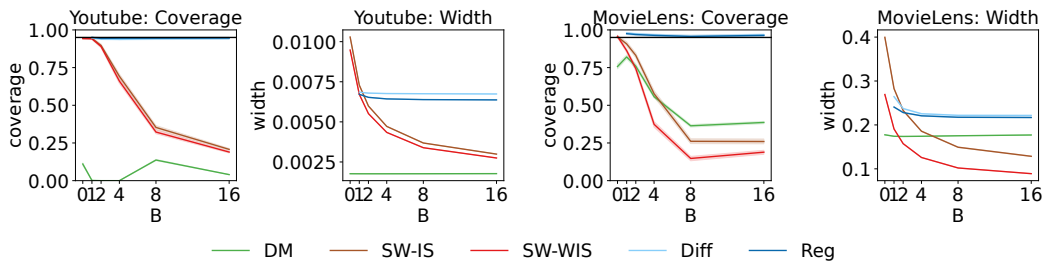


Figure 6.2: The empirical coverage and the width of CIs. Higher coverage and lower width is better.

divided by the total number of rounds K. The results shown here are with $n = 1.0|\mathcal{S}|$. IS ($B = 0$), WIS ($B = 0$), Diff and Reg all have the desired coverage and Reg has the smallest width. All sliding window estimators have large bias when $B > 0$, so the coverage is poor and it is unclear how to compare to estimators with the desired coverage. Note that even with a small value of B, for example, $B = 1$ in MovieLens, sliding window estimators fail to provide a valid CI. The result suggests that Reg provides an accurate and tight CI.

Empirical investigation of the feature vectors. Besides using one past reward prediction as the only feature, we also investigate the utility when we (1) include the context features; and (2) include separate past reward predictions, that is, $\phi_k(s, a) = (1, \hat{r}_{k-B}(s, a), \ldots, \hat{r}_{k-1}(s, a))$ where we learn a reward model $\hat{r}_t$ for interval t from data D_t separately. Since these additional features could be correlated, we use ridge regression when estimating the coefficients. The regularization parameter is chosen by cross-validation.

We aim to answer two questions: (1) whether including the context feature or the past reward predictions helps, and (2) how we should include the past reward information. To answer the questions, we test five different feature vectors: (a) Reg: we use one past reward prediction as described in Section 6.4.2, with and without the context features, (b) Reg-AR: we use separate past reward predictions with and without context features, and (c) Reg-Feature: we use context features only. We show the comparison in Figure 6.3. We find that including the context features helps in general, however, using only the context features is not sufficient. The past reward information helps deal

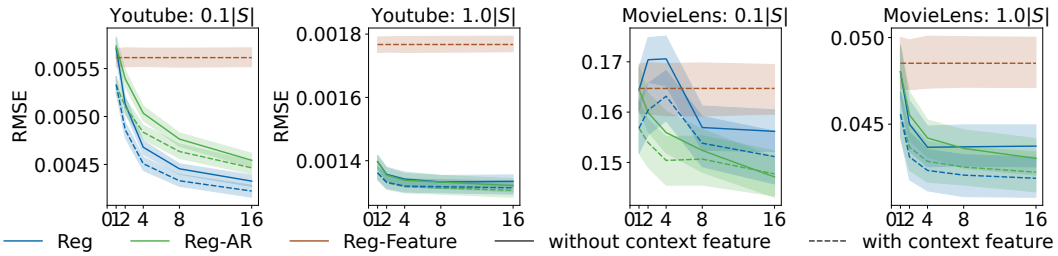


Figure 6.3: Comparison of different feature vectors for estimating $J_k(\pi)$.

with nonstationarity. Moreover, using separate predictions only improves the accuracy in MovieLens with $n = 0.1|\mathcal{S}|$. In these experiments, there was no one dominant way to include past reward information, and more experimentation is needed to understand when one might be preferred.

Ablation study: estimating the population total of the proxy values. We provide an ablation study to investigate the impact when the population total of the proxy value is estimated. For the previous experiments, we use the population total of the proxy values for the DM, Diff and Reg estimator. In this experiment, we test the regression-assisted estimator when the population total of the proxy values are being estimated, that results in the estimator using Eq (6.2), which we call RegDR, and the estimator with an independent survey D', described in the last paragraph of Section 6.4.1, which we call RegDR2.

In Figure 6.4, we found that RegDR2 has a similar RMSE compared to Reg, and both are slightly better than RegDR. This suggests that using the past data to estimate the population total results in very similar performance as we know the population total. Moreover, using an independent survey has potential to improve the standard DR estimator in the non-stationary setting.

6.7 Extension to RL

The estimators for contextual bandits can be extended to finite horizon RL. Let $M = (\mathcal{S}, \mathcal{A}, P, r, H, \nu)$ be a finite horizon finite MDP. Our goal is to estimate the value of a policy $J(\pi) = \sum_{\tau \in (\mathcal{S}\times\mathcal{A})^H} \mathbb{P}^\pi_M(\tau) R(\tau)$ where $\mathbb{P}^\pi_M(\tau) = \nu(s_0)\pi(a_0|s_0)P(s_1|s_0,a_0)\ldots\pi(a_{H-1}|s_{H-1})$ is the probability of see-

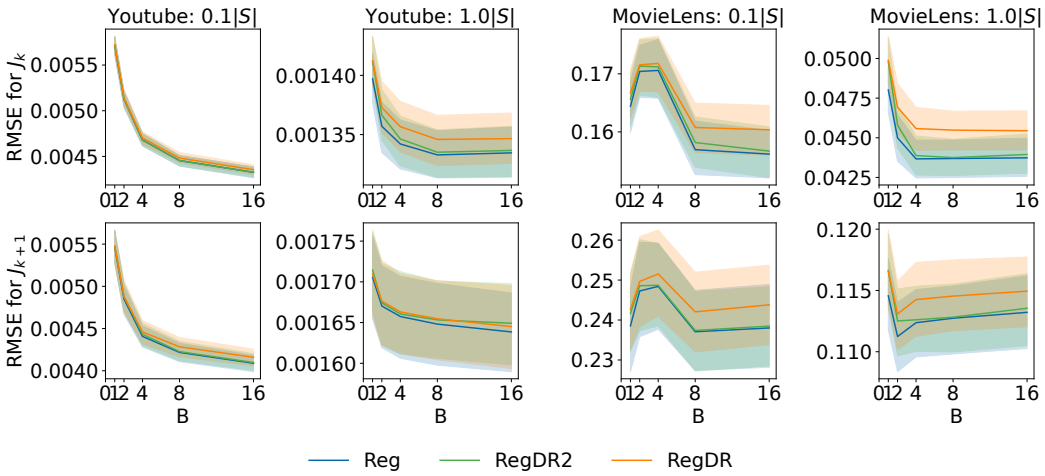


Figure 6.4: Comparison when the population total of the proxy value is estimated. **Top: estimating $J_k(\pi)$. Bottom: predicting $J_{k+1}(\pi)$.**

ing the trajectory $\tau = (s_0, a_0, \ldots, s_{H-1}, a_{H-1})$ by running π in M, and $R(\tau) = \sum_{h=0}^{H-1} r(s_h, a_h)$.

To formalize OPE for RL under survey sampling, let $\mathcal{U} = (\mathcal{S} \times \mathcal{A})^H$ be the population containing all trajectories and $y_\tau = \mathbb{P}_M^\pi(\tau) R(\tau)$ be the study variable. The corresponding estimator is the trajectory-wise IS estimator:

$$\hat{t}_{IS} = \frac{1}{n} \sum_{\tau \in D} \frac{\mathbb{P}_M^\pi(\tau)}{\mathbb{P}_M^{\pi_b}(\tau)} R(\tau) = \frac{1}{n} \sum_{\tau \in D} \prod_{h=0}^{H-1} \frac{\pi(a_h|s_h)}{\pi_b(a_h|s_h)} R(\tau).$$

Note that there are many ways to view OPE for RL in the survey sampling framework, which corresponds to different existing estimators for RL such as the per-decision IS (PDIS) estimator and marginalized IS estimator, but we will focus on the trajectory IS estimator in this section.

The regression-assisted estimator with FQE. We use FQE, which has been shown to be effective for stationary OPE benchmarks empirically (Voloshin et al., 2021), to build a proxy value $\hat{R}(\tau)$ for each trajectories $\tau \in \mathcal{U}$.

In non-stationary environments, FQE outputs $\hat{Q}_{k-1}(s, a)$ from the past data $D_{k-B}, \ldots, D_{k-1}$, and we use $\hat{R}(\tau) = \hat{V}_{k-1}(s_0) = \sum_{a \in \mathcal{A}} \pi(a|s_0) \hat{Q}_{k-1}(s_0, a)$ as the proxy value where s_0 is the initial state of the trajectory τ. Similar to the estimator for contextual bandits, we first estimate the coefficient with the feature vector $\phi(s_0)^\top = (1, \hat{V}_{k-1}(s_0))$ and use the regression-assisted DR

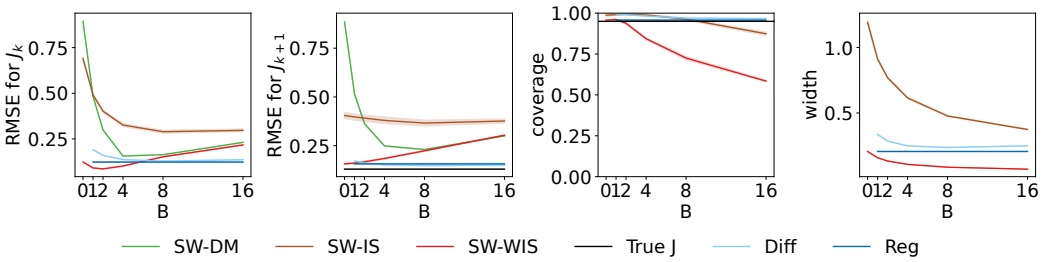


Figure 6.5: Results for the RL environment. **First column: estimating $J_k(\pi)$. Second column: predicting $J_{k+1}(\pi)$. Third and fourth column: coverage and width of CI**.

estimator

$$\hat{t}_{Reg\text{-}FQE,k} = \sum_{s_0 \in \mathcal{S}} \nu(s_0)\phi(s_0)^\top \hat{\beta}_k + \frac{1}{n} \sum_{\tau \in D_k} \prod_{h=0}^{H-1} \frac{\pi(a_h|s_h)}{\pi_b(a_h|s_h)} (R(\tau) - \phi(s_0)^\top \hat{\beta}_k).$$

When ν is unknown, we can estimate the population total of the proxy value by $1/|D'| \sum_{s_0 \in D'} \phi(s_0)^\top \hat{\beta}$ from the past data D' or the same data D_k. The regression-assisted DR estimator can be viewed as a biased-corrected FQE estimator for non-stationary environments.

Experimental results. We consider an RL environment with a binary tree structure, that is, a finite horizon MDP with $H = 10$, $|\mathcal{A}| = 2$, $|\mathcal{S}| = |\mathcal{A}|^H - 1$, and an initial state s_0. For each state, taking action 1 leads to the left child and taking action 2 leads to the right child. The reward for each state-action pair follows a sine wave with different frequency and amplitude. The environment mimics the session-aware recommendation problem where we take a sequence of actions for one customer during a short session. We use a random policy to collect 10 trajectories for every interval. The target policy is a trained policy using Q-learning on the underlying environment.

From Figure 6.5, Reg has the lowest error for estimating the current value and predicting the future value in general. We show the coverage of the one-sided CI since we mainly care about the lower bound on the policy value for safe policy improvement. The results show that Reg again provides a valid and tight interval estimation, and is promising for safe policy improvement in non-stationary RL environments.

6.8 Conclusion

We proposed the regression-assisted DR estimator for OPE in the non-stationary setting, inspired by estimators from the survey sampling literature. The estimator incorporates past data into a proxy value without introducing large bias, and uses a regression approach to build a reward prediction well suited for non-stationary environments. As far as we know, these two ideas have not been applied to non-stationary OPE. We theoretically show that we can construct a large sample confidence interval and empirically demonstrate that the proposed estimator provides tight and valid high-confidence estimation in several recommendation environments in contextual bandits and finite horizon RL.

Chapter 7

Conclusion

This thesis aims to bridge the gap between theory and practice, ultimately paving the way for real-world offline RL systems. Our objective is to enhance our understanding of the practical aspects of offline RL, and to develop sample efficient algorithms with provable performance guarantees.

We first studied the policy selection problem in the offline setting. We established a clear and fundamental connection between OPE and OPS, highlighting that OPE represents the optimal approach for OPS in the worst case scenarios. We then proposed a sample efficient OPS method based on BE estimation, and explored the scenarios in which BE has advantages over OPE for OPS. The findings from this chapter provide a better understanding on when OPS can be sample efficient, and valuable insights for future development of sample efficient OPS methods.

We then introduced an MDP property, called Action Impact Regularity. Building upon this concept, an FQI-based algorithm, called FQI-AIR, was developed for MDPs satisfying AIR. We proved that the algorithm can return a nearly optimal policy with polynomial sample complexity, without relying on impractical data coverage assumptions. Furthermore, extensive empirical evaluations were performed, demonstrating the superior performance of the proposed algorithm compared to existing offline RL methods that do not exploit the property.

The focus shifted towards exploring offline RL in non-stationary environments, which are prevalent in real-world applications. To address this chal-

lenging scenario, we proposed the regression-assisted doubly robust estimator to leverage past data without introducing large bias. The theoretical analysis demonstrated the construction of large sample confidence intervals, ensuring reliable high-confidence estimation. Empirical investigations were carried out in various recommendation environments, affirming the effectiveness of the proposed estimator, which provides tight and valid high-confidence estimates.

7.1 Limitations and future directions

Query efficiency when learning with simulators. The FQI-AIR algorithm presented in Chapter 5 randomly samples the endogenous state and action to generate synthetic transitions. However, the uniform sampling might not be effective for large endogenous state space and action space. There are several efficient and theoretically sound approaches under linear function approximation (Lattimore et al., 2020; Shariff & Szepesvari, 2020). It is an interesting direction to extend such ideas to design effective sampling distribution with general function approximation.

More generally, consider the setting that we have a simulator of the underlying environment that we can query the reward and the next state from some state-action pairs. For example, we might be able to query the state-action pairs that the agent has visited in the past, which is called the local access protocol (D. Yin et al., 2022). A natural question is to study how we can better use the local access simulator to find a good policy in terms of query efficiency.

Non-stationary OPE for long horizon tasks. We proposed a regression-assisted estimator for short horizon RL tasks in Chapter 6. However, the estimator still uses the product of IS ratios, so it would have high variance for long horizon tasks. An important direction is to design estimators that work for long horizon tasks and have small bias from reusing past data.

Beyond fully offline setting: hybrid RL. In this thesis, we mainly focus on the setting where we learn purely from an offline data. In some applica-

tions, we might be able to perform a certain amount of online interactions or simulation, which could greatly improve the applicability of RL. The setting is called *hybrid RL* (Song et al., 2023).

In Chapter 4, we highlighted the difficulty of OPS in the fully offline setting. If we can perform online iteration or simulations (similar to online AB testing), we would be able to evaluate candidate policies even though the offline data does not provide full coverage. There are several important research questions toward this direction; for example, how can we design better OPS approachs that leverage the offline data and a budget of online interactions? We can also consider using offline data to design safe data collection policies to improve data coverage.

References

Abdullah, H. M., Gastli, A., & Ben-Brahim, L. (2021). Reinforcement learning based EV charging management systems–a review. *IEEE Access*.

Abernethy, J., & Kale, S. (2013). Adaptive market making via online learning. *Advances in Neural Information Processing Systems*.

Agarwal, A., Jiang, N., Kakade, S. M., & Sun, W. (2019). Reinforcement learning: Theory and algorithms.

Agarwal, A., Kakade, S., & Yang, L. F. (2020). Model-based reinforcement learning with a generative model is minimax optimal. *Conference on Learning Theory*.

Agarwal, R., Schuurmans, D., & Norouzi, M. (2020). An optimistic perspective on offline reinforcement learning. *International Conference on Machine Learning*.

Antos, A., Szepesvári, C., & Munos, R. (2008). Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*.

Babaioff, M., Immorlica, N., Kempe, D., & Kleinberg, R. (2007). A knapsack secretary problem with applications. In *Approximation, randomization, and combinatorial optimization. algorithms and techniques*. Springer.

Bartlett, P. L., Boucheron, S., & Lugosi, G. (2002). Model selection and error estimation. *Machine Learning*.

Bellemare, M. G., Veness, J., & Bowling, M. (2012). Investigating contingency awareness using Atari 2600 games. *AAAI Conference on Artificial Intelligence*.

Bertsimas, D., & Lo, A. W. (1998). Optimal control of execution costs. *Journal of Financial Markets*.

Bottou, L., Peters, J., Quiñonero-Candela, J., Charles, D. X., Chickering, D. M., Portugaly, E., Ray, D., Simard, P., & Snelson, E. (2013). Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research*.

Brafman, R. I., & Tennenholtz, M. (2002). R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*.

Breidt, F. J., & Opsomer, J. D. (2000). Local polynomial regression estimators in survey sampling. *Annals of statistics*.

Burhani, H., Ding, G. W., Hernandez-Leal, P., Prince, S., Shi, D., & Szeto, S. (2020). Aiden – reinforcement learning for order execution [Online; accessed 1-June-2022].

Cassel, C. M., Särndal, C. E., & Wretman, J. H. (1976). Some results on generalized difference estimation and generalized regression estimation for finite populations. *Biometrika*.

Chambers, R., & Clark, R. (2012). *An introduction to model-based survey sampling with applications*. Oxford University Press.

Chandak, Y., Niekum, S., da Silva, B., Learned-Miller, E., Brunskill, E., & Thomas, P. S. (2021). Universal off-policy evaluation. *Advances in Neural Information Processing Systems*.

Chandak, Y., Theocharous, G., Shankar, S., White, M., Mahadevan, S., & Thomas, P. (2020). Optimizing for the future in non-stationary MDPs. *International Conference on Machine Learning*.

Chang, J., Wang, K., Kallus, N., & Sun, W. (2022). Learning Bellman complete representations for offline policy evaluation. *International Conference on Machine Learning*.

Chen, J., & Jiang, N. (2019). Information-theoretic considerations in batch reinforcement learning. *International Conference on Machine Learning*.

Cheng, C.-A., Xie, T., Jiang, N., & Agarwal, A. (2022). Adversarially trained actor critic for offline reinforcement learning. *International Conference on Machine Learning*.

Chitnis, R., & Lozano-Pérez, T. (2020). Learning compact models for planning with exogenous processes. *Conference on Robot Learning*.

Cochran, W. G. (1977). *Sampling techniques, 3rd edition.* John Wiley.

Dai, B., Shaw, A., Li, L., Xiao, L., He, N., Liu, Z., Chen, J., & Song, L. (2018). SBEED: Convergent reinforcement learning with nonlinear function approximation. *International Conference on Machine Learning*.

Dietterich, T., Trimponias, G., & Chen, Z. (2018). Discovering and removing exogenous state variables and rewards for reinforcement learning. *International Conference on Machine Learning*.

Doroudi, S., Thomas, P. S., & Brunskill, E. (2017). Importance sampling for fair policy selection. *Conference on Uncertainty in Artificial Intelligence*.

Duan, Y., Jin, C., & Li, Z. (2021). Risk bounds and Rademacher complexity in batch reinforcement learning. *International Conference on Machine Learning*.

Dudik, M., Erhan, D., Langford, J., & Li, L. (2012). Sample-efficient nonstationary policy evaluation for contextual bandits. *Conference on Uncertainty in Artificial Intelligence*.

Dudik, M., Langford, J., & Li, L. (2011). Doubly robust policy evaluation and learning. *International Conference on Machine Learning*.

Efroni, Y., Foster, D. J., Misra, D., Krishnamurthy, A., & Langford, J. (2022). Sample-efficient reinforcement learning in the presence of exogenous information. *Conference on Learning Theory*.

Efroni, Y., Misra, D., Krishnamurthy, A., Agarwal, A., & Langford, J. (2021). Provable RL with exogenous distractors via multistep inverse dynamics. *arXiv preprint arXiv:2110.08847*.

Ernst, D., Geurts, P., & Wehenkel, L. (2005). Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*.

Fan, J., Wang, Z., Xie, Y., & Yang, Z. (2020). A theoretical analysis of deep Q-learning. *Learning for dynamics and control*.

Farahmand, A.-m., Munos, R., & Szepesvári, C. (2010). Error propagation for approximate policy and value iteration. *Advances in Neural Information Processing Systems*.

Farahmand, A.-m., & Szepesvári, C. (2011). Model selection in reinforcement learning. *Machine Learning*.

Farajtabar, M., Chow, Y., & Ghavamzadeh, M. (2018). More robust doubly robust off-policy evaluation. *International Conference on Machine Learning*.

Félix-Medina, M. H. (2003). Asymptotics in adaptive cluster sampling. *Environmental and Ecological Statistics*.

Foster, D. J., Krishnamurthy, A., & Luo, H. (2019). Model selection for contextual bandits. *Advances in Neural Information Processing Systems*.

Foster, D. J., Krishnamurthy, A., Simchi-Levi, D., & Xu, Y. (2021). Offline reinforcement learning: Fundamental barriers for value function approximation. *arXiv preprint arXiv:2111.10919*.

Freeman, P. (1983). The secretary problem and its extensions: A review. *International Statistical Review/Revue Internationale de Statistique*.

Fu, J., Kumar, A., Nachum, O., Tucker, G., & Levine, S. (2020). D4RL: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*.

Fujimoto, S., & Gu, S. S. (2021). A minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems*.

Fujimoto, S., Meger, D., & Precup, D. (2019). Off-policy deep reinforcement learning without exploration. *International Conference on Machine Learning*.

Goldstein, D. G., McAfee, R. P., Suri, S., & Wright, J. R. (2020). Learning when to stop searching. *Management Science*.

Gulcehre, C., Wang, Z., Novikov, A., Paine, T., Gómez, S., Zolna, K., Agarwal, R., Merel, J. S., Mankowitz, D. J., Paduraru, C., et al. (2020). RL unplugged: A suite of benchmarks for offline reinforcement learning. *Advances in Neural Information Processing Systems*.

Hallak, A., Di-Castro, D., & Mannor, S. (2013). Model selection in markovian processes. *International Conference on Knowledge Discovery and Data Mining*.

Hammersley, J. M., & Handscomb, D. C. (1964). *Monte carlo methods*. Springer Dordrecht.

Hansen, M. H., & Hurwitz, W. N. (1943). On the theory of sampling from finite populations. *The Annals of Mathematical Statistics*.

Harper, F. M., & Konstan, J. A. (2015). The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*.

Hong, J., Kveton, B., Zaheer, M., Chow, Y., & Ahmed, A. (2021). Non-stationary off-policy optimization. *International Conference on Artificial Intelligence and Statistics*.

Horvitz, D. G., & Thompson, D. J. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*.

Jagerman, R., Markov, I., & de Rijke, M. (2019). When people change their mind: Off-policy evaluation in non-stationary recommendation environments. *International Conference on Web Search and Data Mining*.

Jiang, N. (2018). PAC reinforcement learning with an imperfect model. *AAAI Conference on Artificial Intelligence*.

Jiang, N., & Li, L. (2016). Doubly robust off-policy value evaluation for reinforcement learning. *International Conference on Machine Learning*.

Jin, C., Yang, Z., Wang, Z., & Jordan, M. I. (2020). Provably efficient reinforcement learning with linear function approximation. *Conference on Learning Theory*.

Kakade, S. M. (2003). *On the sample complexity of reinforcement learning* (Doctoral dissertation). University College London.

Kallus, N., & Uehara, M. (2019). Intrinsically efficient, stable, and bounded off-policy evaluation for reinforcement learning. *Advances in Neural Information Processing Systems*.

Kearns, M., & Singh, S. (2002). Near-optimal reinforcement learning in polynomial time. *Machine Learning*.

Kidambi, R., Rajeswaran, A., Netrapalli, P., & Joachims, T. (2020). Morel: Model-based offline reinforcement learning. *Advances in Neural Information Processing Systems*.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kostrikov, I., Nair, A., & Levine, S. (2022). Offline reinforcement learning with implicit Q-learning. *International Conference on Learning Representations*.

Krishnamurthy, A., Agarwal, A., & Langford, J. (2016). PAC reinforcement learning with rich observations. *Advances in Neural Information Processing Systems*.

Kumar, A., Fu, J., Soh, M., Tucker, G., & Levine, S. (2019). Stabilizing off-policy Q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*.

Kumar, A., Singh, A., Tian, S., Finn, C., & Levine, S. (2021). A workflow for offline model-free robotic reinforcement learning. *Conference on Robot Learning*.

Kumar, A., Zhou, A., Tucker, G., & Levine, S. (2020). Conservative Q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*.

Kunnumkal, S., & Topaloglu, H. (2008). Using stochastic approximation methods to compute optimal base-stock levels in inventory control problems. *Operations Research*.

Kuzborskij, I., Vernade, C., Gyorgy, A., & Szepesvári, C. (2021). Confident off-policy evaluation and selection through self-normalized importance weighting. *International Conference on Artificial Intelligence and Statistics*.

Laroche, R., Trichelair, P., & Des Combes, R. T. (2019). Safe policy improvement with baseline bootstrapping. *International Conference on Machine Learning*.

Lattimore, T., Szepesvari, C., & Weisz, G. (2020). Learning with good feature representations in bandits and in RL with a generative model. *International Conference on Machine Learning*.

Lattimore, T., & Szepesvári, C. (2020). *Bandit algorithms*. Cambridge University Press.

Lazaric, A., Restelli, M., & Bonarini, A. (2008). Transfer of samples in batch reinforcement learning. *International Conference on Machine Learning*.

Le, H., Voloshin, C., & Yue, Y. (2019). Batch policy learning under constraints. *International Conference on Machine Learning*.

Lee, J., Pacchiano, A., Muthukumar, V., Kong, W., & Brunskill, E. (2021). Online model selection for reinforcement learning with function approximation. *International Conference on Artificial Intelligence and Statistics*.

Lee, J., Tucker, G., Nachum, O., & Dai, B. (2022). Model selection in batch policy optimization. *International Conference on Machine Learning*.

Lee, J. N., Tucker, G., Nachum, O., Dai, B., & Brunskill, E. (2022). Oracle inequalities for model selection in offline reinforcement learning. *Advances in Neural Information Processing Systems*.

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

Li, L., Chu, W., Langford, J., & Wang, X. (2011). Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. *International Conference on Web Search and Data Mining*.

Lian, R., Peng, J., Wu, Y., Tan, H., & Zhang, H. (2020). Rule-interposing deep reinforcement learning based energy management strategy for power-split hybrid electric vehicle. *Energy*.

Liu, Q., Li, L., Tang, Z., & Zhou, D. (2018). Breaking the curse of horizon: Infinite-horizon off-policy estimation. *Advances in Neural Information Processing Systems*.

Liu, V., Chandak, Y., Thomas, P., & White, M. (2023). Asymptotically unbiased off-policy policy evaluation when reusing old data in nonstationary environments. *International Conference on Artificial Intelligence and Statistics*.

Liu, V., Wright, J. R., & White, M. (2023). Exploiting action impact regularity and exogenous state variables for offline reinforcement learning. *Journal of Artificial Intelligence Research*.

Liu, Y., Swaminathan, A., Agarwal, A., & Brunskill, E. (2020). Provably good batch reinforcement learning without great exploration. *Advances in Neural Information Processing Systems*.

Machado, M. C., Bellemare, M. G., Talvitie, E., Veness, J., Hausknecht, M., & Bowling, M. (2018). Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research*.

Mahmood, A. R., Van Hasselt, H. P., & Sutton, R. S. (2014). Weighted importance sampling for off-policy learning with linear function approximation. *Advances in Neural Information Processing Systems*.

Mannor, S., Simester, D., Sun, P., & Tsitsiklis, J. N. (2007). Bias and variance approximation in value function estimates. *Management Science*.

Maurer, A., & Pontil, M. (2009). Empirical bernstein bounds and sample variance penalization. *Conference on Learning Theory*.

McConville, K. (2011). *Improved estimation for complex surveys using modern regression techniques* (Doctoral dissertation). Colorado State University.

McConville, K. S., Breidt, F. J., Lee, T., & Moisen, G. G. (2017). Model-assisted survey regression estimation with the lasso. *Journal of Survey Statistics and Methodology*.

McGregor, S., Houtman, R., Montgomery, C., Metoyer, R., & Dietterich, T. G. (2017). Factoring exogenous state for model-free Monte Carlo. *arXiv preprint arXiv:1703.09390*.

Meuleau, N., Hauskrecht, M., Kim, K.-E., Peshkin, L., Kaelbling, L. P., Dean, T. L., & Boutilier, C. (1998). Solving very large weakly coupled markov decision processes. *AAAI Conference on Artificial Intelligence*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*.

Munos, R. (2003). Error bounds for approximate policy iteration. *International Conference on Machine Learning*.

Munos, R. (2005). Error bounds for approximate value iteration. *National Conference on Artificial Intelligence*.

Munos, R. (2007). Performance bounds in l_p-norm for approximate value iteration. *SIAM Journal on Control and Optimization*.

Nachum, O., Chow, Y., Dai, B., & Li, L. (2019). DualDICE: Behavior-agnostic estimation of discounted stationary distribution corrections. *Advances in Neural Information Processing Systems*.

Nevmyvaka, Y., Feng, Y., & Kearns, M. (2006). Reinforcement learning for optimized trade execution. *International Conference on Machine Learning*.

Nie, A., Flet-Berliac, Y., Jordan, D. R., Steenbergen, W., & Brunskill, E. (2022). Data-efficient pipeline for offline reinforcement learning with limited data. *Advances in Neural Information Processing Systems*.

Paine, T. L., Paduraru, C., Michi, A., Gulcehre, C., Zolna, K., Novikov, A., Wang, Z., & de Freitas, N. (2020). Hyperparameter selection for offline reinforcement learning. *arXiv preprint arXiv:2007.09055*.

Patterson, A., White, A., & White, M. (2022). A generalized projected Bellman error for off-policy value estimation in reinforcement learning. *Journal of Machine Learning Research*.

Precup, D., Sutton, R. S., & Singh, S. P. (2000). Eligibility traces for off-policy policy evaluation. *International Conference on Machine Learning*.

Puterman, M. L. (2014). *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons.

Qi, H., Su, Y., Kumar, A., & Levine, S. (2022). Data-driven offline decision-making via invariant representation learning. *Advances in Neural Information Processing Systems*.

Rashidinejad, P., Zhu, B., Ma, C., Jiao, J., & Russell, S. (2021). Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information Processing Systems*.

Rubinstein, R. Y. (1981). *Simulation and the monte carlo method*. John Wiley & Sons.

Sachdeva, N., Su, Y., & Joachims, T. (2020). Off-policy bandits with deficient support. *International Conference on Knowledge Discovery & Data Mining*.

Särndal, C.-E., Swensson, B., & Wretman, J. (1992). *Model assisted survey sampling*. Springer New York.

Särndal, C.-E., Swensson, B., & Wretman, J. H. (1989). The weighted residual technique for estimating the variance of the general regression estimator of the finite population total. *Biometrika*.

Seno, T., & Imai, M. (2022). D3rlpy: An offline deep reinforcement learning library. *Journal of Machine Learning Research*.

Shahamiri, M. (2008). *Reinforcement learning in environments with independent delayed-sense dynamics* (Master's thesis). University of Alberta.

Shariff, R., & Szepesvari, C. (2020). Efficient planning in large mdps with weak linear function approximation. *Advances in Neural Information Processing Systems*.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *Nature*.

Song, Y., Zhou, Y., Sekhari, A., Bagnell, J. A., Krishnamurthy, A., & Sun, W. (2023). Hybrid RL: Using both offline and online data can make RL efficient. *International Conference on Learning Representations*.

Su, Y., Dimakopoulou, M., Krishnamurthy, A., & Dudik, M. (2020). Doubly robust off-policy evaluation with shrinkage. *International Conference on Machine Learning*.

Su, Y., Srinath, P., & Krishnamurthy, A. (2020). Adaptive estimator selection for off-policy evaluation. *International Conference on Machine Learning*.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Swaminathan, A., & Joachims, T. (2015a). Counterfactual risk minimization: Learning from logged bandit feedback. *International Conference on Machine Learning*.

Swaminathan, A., & Joachims, T. (2015b). The self-normalized estimator for counterfactual learning. *Advances in Neural Information Processing Systems*.

Tang, S., & Wiens, J. (2021). Model selection for offline reinforcement learning: Practical considerations for healthcare settings. *Machine Learning for Healthcare Conference*.

Thomas, P., & Brunskill, E. (2016). Data-efficient off-policy policy evaluation for reinforcement learning. *International Conference on Machine Learning*.

Thomas, P., Theocharous, G., & Ghavamzadeh, M. (2015a). High confidence policy improvement. *International Conference on Machine Learning*.

Thomas, P., Theocharous, G., & Ghavamzadeh, M. (2015b). High-confidence off-policy evaluation. *AAAI Conference on Artificial Intelligence*.

Thomas, P. S., Theocharous, G., Ghavamzadeh, M., Durugkar, I., & Brunskill, E. (2017). Predictive off-policy policy evaluation for nonstationary decision problems, with applications to digital marketing. *IAAI Conference*.

Trabucco, B., Kumar, A., Geng, X., & Levine, S. (2021). Conservative objective models for effective offline model-based optimization. *International Conference on Machine Learning*.

Uehara, M., Huang, J., & Jiang, N. (2020). Minimax weight and q-function learning for off-policy evaluation. *International Conference on Machine Learning*.

Uehara, M., & Sun, W. (2022). Pessimistic model-based offline reinforcement learning under partial coverage. *International Conference on Learning Representations*.

Van Roy, B., Bertsekas, D. P., Lee, Y., & Tsitsiklis, J. N. (1997). A neuro-dynamic programming approach to retailer inventory management. *IEEE Conference on Decision and Control*.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*.

Voloshin, C., Le, H. M., Jiang, N., & Yue, Y. (2021). Empirical study of off-policy policy evaluation for reinforcement learning. *Neural Information Processing Systems Datasets and Benchmarks Track*.

Wang, R., Foster, D. P., & Kakade, S. M. (2021). What are the statistical limits of offline RL with linear function approximation? *International Conference on Learning Representations*.

Wang, Y.-X., Agarwal, A., & Dudık, M. (2017). Optimal and adaptive off-policy evaluation in contextual bandits. *International Conference on Machine Learning*.

Wu, Y., Tucker, G., & Nachum, O. (2019). Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*.

Xiao, C., Lee, I., Dai, B., Schuurmans, D., & Szepesvari, C. (2022). The curse of passive data collection in batch reinforcement learning. *International Conference on Artificial Intelligence and Statistics*.

Xie, T., Cheng, C.-A., Jiang, N., Mineiro, P., & Agarwal, A. (2021). Bellman-consistent pessimism for offline reinforcement learning. *Advances in Neural Information Processing Systems*.

Xie, T., & Jiang, N. (2020). Q* approximation schemes for batch reinforcement learning: A theoretical comparison. *Conference on Uncertainty in Artificial Intelligence*.

Xie, T., & Jiang, N. (2021). Batch value-function approximation with only realizability. *International Conference on Machine Learning*.

Xie, T., Ma, Y., & Wang, Y.-X. (2019). Towards optimal off-policy evaluation for reinforcement learning with marginalized importance sampling. *Advances in Neural Information Processing Systems*.

Yang, M., Dai, B., Nachum, O., Tucker, G., & Schuurmans, D. (2022). Offline policy selection under uncertainty. *International Conference on Artificial Intelligence and Statistics*.

Yang, S., & Kim, J. K. (2020). Statistical data integration in survey sampling: A review. *Japanese Journal of Statistics and Data Science*.

Yin, D., Hao, B., Abbasi-Yadkori, Y., Lazić, N., & Szepesvári, C. (2022). Efficient local planning with linear function approximation. *International Conference on Algorithmic Learning Theory*.

Yin, M., Bai, Y., & Wang, Y.-X. (2021). Near-optimal provable uniform convergence in offline policy evaluation for reinforcement learning. *International Conference on Artificial Intelligence and Statistics*.

Yin, M., & Wang, Y.-X. (2021). Optimal uniform OPE and model-based offline reinforcement learning in time-homogeneous, reward-free and task-agnostic settings. *Advances in Neural Information Processing Systems*.

Yu, J. Y., & Mannor, S. (2009). Piecewise-stationary bandit problems with side observations. *International Conference on Machine Learning*.

Yu, T., Kumar, A., Rafailov, R., Rajeswaran, A., Levine, S., & Finn, C. (2021). Combo: Conservative offline model-based policy optimization. *Advances in Neural Information Processing Systems*.

Zanette, A. (2020). Exponential lower bounds for batch reinforcement learning: Batch RL can be exponentially harder than online RL. *arXiv preprint arXiv:2012.08005*.

Zanette, A. (2022). When is realizability sufficient for off-policy reinforcement learning? *arXiv preprint arXiv:2211.05311*.

Zhan, W., Huang, B., Huang, A., Jiang, N., & Lee, J. (2022). Offline reinforcement learning with realizability and single-policy concentrability. *Conference on Learning Theory*.

Zhang, S., & Jiang, N. (2021). Towards hyperparameter-free policy selection for offline reinforcement learning. *Advances in Neural Information Processing Systems*.

Zitovsky, J. P., De Marchi, D., Agarwal, R., & Kosorok, M. R. (2023). Revisiting Bellman errors for offline model selection. *International Conference on Machine Learning*.

Appendix A

Supplementary Material for OPS

We provide supplementary material for Chapter 4.

A.1 Theoretical analysis

A.1.1 Proof of Theorem 1

Theorem 1 (Upper bound on sample complexity of OPS)**.** Given an MDP M, a data distribution d_b, and a set of policies Π, suppose that, for any pair (ε, δ), there exists an (ε, δ)-sound OPE algorithm $\mathcal{L}$ on any OPE instance $I \in \{(M, d_b, \pi) : \pi \in \Pi\}$ with a sample size at most $O(\mathrm{N}_{OPE}(S, A, H, 1/\varepsilon, 1/\delta))$. Then there exists an (ε, δ)-sound OPS algorithm for the OPS problem instance (M, d_b, Π) which requires at most $O(\mathrm{N}_{OPE}(S, A, H, 2/\varepsilon, |\Pi|/\delta))$ episodes.

Proof. The OPS algorithm $\mathcal{L}(D, \Pi)$ for a given (ε, δ) works as follows: we query an (ε', δ')-sound OPE algorithm for each policy in Π and select the policy with the highest estimated value. That is, $\mathcal{L}(D, \Pi)$ outputs the policy $\bar{\pi} := \arg\max_{\pi \in \Pi} \hat{J}(\pi)$, where $\hat{J}(\pi)$ is the value estimate for policy π by the (ε', δ')-sound OPE algorithm with data D.

By definition of an (ε', δ')-sound OPE algorithm we have

$$\Pr_{D \sim d_b}(|\hat{J}(\pi) - J(\pi)| \leq \varepsilon') \geq 1 - \delta', \forall \pi \in \Pi.$$

Applying the union bound, we have

$$\Pr_{D \sim d_b}(\forall \pi \in \Pi, |\hat{J}(\pi) - J(\pi)| \leq \varepsilon') \geq 1 - \delta'|\Pi|.$$

Let $\pi^\dagger$ denote the best policy in the candidate set Π, that is, $\pi^\dagger := \arg\max_{\pi\in\Pi} J(\pi)$. With probability $1 - \delta'|\Pi|$, we have

$$J(\bar{\pi}) \geq \hat{J}(\bar{\pi}) - \epsilon' \geq \hat{J}(\pi^\dagger) - \varepsilon' \geq J(\pi^\dagger) - 2\varepsilon'.$$

The second inequality follows from the definition of $\bar{\pi}$. Finally, by setting $\delta' = \delta/|\Pi|$ and $\varepsilon' = \varepsilon/2$, we get

$$\Pr_{D\sim d_b}\left(J(\bar{\pi}) \geq J(\pi^\dagger) - \varepsilon\right) \geq 1 - \delta.$$

That is, $\mathcal{L}$ is an (ε, δ)-sound OPS algorithm, and it requires at most $O(\mathrm{N}_{OPE}(S, A, H, 2/\varepsilon, |\Pi|/\delta))$ samples. □

A.1.2 Proof of Theorem 2

Theorem 2 (Lower bound on sample complexity of OPS)**.** Suppose for any data distribution d_b and any pair (ε, δ) with $\varepsilon \in (0, V_{max}/2)$ and $\delta \in (0, 1)$, there exists an MDP M and a policy π such that any (ε, δ)-sound OPE algorithm requires at least $\Omega(\mathrm{N}_{OPE}(S, A, H, 1/\varepsilon, 1/\delta))$ episodes. Then there exists an MDP M' with $S' = S + 2$, $H' = H + 1$, and a set of candidate policies such that for any pair (ε, δ) with $\varepsilon \in (0, V_{max}/3)$ and $\delta \in (0, 1/m)$ where $m := \lceil \log(V_{max}/\varepsilon) \rceil \geq 1$, any (ε, δ)-sound OPS algorithm also requires at least $\Omega(\mathrm{N}_{OPE}(S, A, H, 3/2\varepsilon, 1/m\delta))$ episodes.

Proof. Our goal is to construct an (ε, δ)-sound OPE algorithm with $\delta \in (0, 1)$ and $\varepsilon \in [0, V_{max}/2]$. To evaluate any policy π in M with dataset D sampled from d_b, we first construct a new MDP M_r with two additional states: an initial state s_0 and a terminal state s_1. Taking a_1 at s_0 transitions to s_1 with reward r. Taking a_2 at s_0 transitions to the initial state in the *original* MDP M. See Figure 4.3 in the main paper for visualization.

Let $\Pi = \{\pi_1, \pi_2\}$ be the candidate set in M_r where $\pi_1(s_0) = a_1$ and $\pi_2(s_0) = a_2$ and $\pi_2(a|s) = \pi(a|s)$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$. Since π_1 always transitions to s_1, it never transitions to states in MDP M. Therefore, π_1 can be arbitrary for all $(s, a) \in \mathcal{S} \times \mathcal{A}$. We can add any number of transitions (s_0, a_1, r, s) and $(s_0, a_2, 0, s)$ in D to construct the dataset D_r with distribution $d_{b,r}$ arbitrarily.

Suppose we have an (ε', δ')-sound OPS algorithm, where we set $\varepsilon' = 2\varepsilon/3$ and $\delta' = \delta/m$ with $m := \lceil \log(V_{max}/\varepsilon') \rceil$. Note that if this assumption does not hold, then it directly implies that the sample complexity of OPS is larger than $\Omega(\mathrm{N}_{OPE}(S, A, H, 1/\varepsilon, 1/\delta))$. Our strategy will be to iteratively set the reward r of MDP M_r and run our sound OPS algorithm on Π and using bisection search to estimate a precise interval for $J(\pi)$.

The process is as follows. By construction, our OPS algorithm will output either π_1, which has value $J_{M_r}(\pi_1) = r$, or output π_2, which has value $J_{M_r}(\pi_2) = J_M(\pi)$. That is, it has the same value as π in the original MDP. Let us consider the following two cases. Let $\pi^\dagger$ be the best policy in Π for MDP M_r.

Case 1: the OPS algorithm selects π_1. We know, by definition of a sound OPS algorithm, that

$$\begin{aligned} & \Pr(J_{M_r}(\pi_1) \geq J_{M_r}(\pi^\dagger) - \varepsilon') \geq 1 - \delta' \\ \implies & \Pr(r \geq \max(r, J_{M_r}(\pi_2)) - \varepsilon') \geq 1 - \delta' \\ \implies & \Pr(J_{M_r}(\pi_2) \leq r + \varepsilon') \geq 1 - \delta'. \end{aligned}$$

Case 2: the OPS algorithm selects π_2.

$$\begin{aligned} & \Pr(J_{M_r}(\pi_2) \geq J_{M_r}(\pi^\dagger) - \varepsilon') \geq 1 - \delta' \\ \implies & \Pr(J_{M_r}(\pi_2) \geq \max(r, J_{M_r}(\pi_2)) - \varepsilon') \geq 1 - \delta' \\ \implies & \Pr(J_{M_r}(\pi_2) \geq r - \varepsilon') \geq 1 - \delta'. \end{aligned}$$

Given this information, we describe the iterative process by which we produce the estimate $\hat{J}(\pi)$. We first set $U = V_{max}, L = 0$ and $r = \frac{U+L}{2}$ and run the sound OPS algorithm with input D_r of sample size n_r and the candidate set Π. Then if the selected policy is π_1, then we conclude the desired event $J(\pi) \leq r + \varepsilon'$ occurs with probability at least $1 - \delta'$, and set U equal to r. If the selected policy is π_2, then we know the desired event $J(\pi) \geq r - \varepsilon'$ occurs with probability at least $1 - \delta'$, and set L equal to r. We can continue the bisection search until the accuracy is less than ε', that is, $U - L \leq \varepsilon'$, and the output value estimate is $\hat{J}(\pi) = \frac{U+L}{2}$.

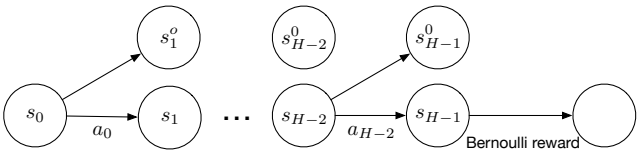


Figure A.1: Lower bound construction.

If all desired events at each call occur, then we conclude that $L - \varepsilon' \leq J(\pi) \leq U + \varepsilon'$ and thus $|J(\pi) - \hat{J}(\pi)| \leq \varepsilon$. The total number of OPS calls is at most m. Setting $\delta' = \delta/m$ and applying a union bound, we can conclude that with probability at least $1 - \delta$, $|J(\pi) - \hat{J}(\pi)| \leq \varepsilon$.

Finally, since any (ε, δ)-sound OPE algorithm on the instance (M, d_b, π) needs at least $\Omega(\mathrm{N}_{OPE}(S, A, H, \varepsilon, 1/\delta))$ samples, the (ε', δ')-sound OPS algorithm also needs at least $\Omega(N_{OPE}(S, A, H, \varepsilon, 1/\delta))$, or equivalently $\Omega(\mathrm{N}_{OPE}(S, A, H, 3/(2\varepsilon'), 1/(m\delta')))$ samples for at least one of the instances $(M_r, d_{b,r}, \Pi)$. □

A.1.3 Proof of Corollary 2

Theorem A.1 (Exponential lower bound on the sample complexity of OPE)**.** For any positive integers S, A, H with $S > 2H$ and a pair (ε, δ) with $0 < \varepsilon \leq \sqrt{1/8}$, $\delta \in (0, 1)$, any (ε, δ)-sound OPE algorithm needs at least $\Omega(A^H \ln(1/\delta)/\varepsilon^2)$ episodes.

Proof. We provide a proof which uses the construction from Xiao et al. (2022). They provide the result for the offline RL problem with Gaussian rewards. Here we provide the result for OPE problem with Bernoulli rewards since we assume rewards are bounded to match Theorem 2.

We can construct an MDP with S states, A actions and $2H$ states. See Figure A.1 for the construction. Given any behavior policy π_b, let $a_h = \arg\min_a \pi_b(a|s_h)$ be the action that leads to the next state s_{h+1} from state s_h, and all other actions lead to an absorbing state s_h^o. Once we reach an absorbing state, the agent gets zero reward for all actions for the remainder of the episode. The only nonzero reward is in the last state s_{H-1}. Consider a target policy that chooses a_h for state s_h for all $h = 0, \ldots, H-1$, and two MDPs

where the only difference between them is the reward distribution in s_{H-1}. MDP 1 has Bernoulli distribution with mean $1/2$ and MDP 2 has Bernoulli distribution with mean $1/2 - 2\varepsilon$. Let $\mathbb{P}_1$ denote the probability measure with respect to MDP 1 and $\mathbb{P}_2$ denote the probability measure with respect to MDP 2.

Let $\hat{r}$ denote the OPE estimate by an algorithm $\mathcal{L}$. Define an event $E = \{\hat{r} < \frac{1}{2} - \varepsilon\}$. Then $\mathcal{L}$ is not (ε, δ)-sound if $(\mathbb{P}_1(E) + \mathbb{P}_2(E^c))/2 \geq \delta$. This is because $\mathcal{L}$ is not (ε, δ)-sound if either $\mathbb{P}_1(\hat{r} < \frac{1}{2} - \varepsilon) \geq \delta$ or $\mathbb{P}_2(\hat{r} > \frac{1}{2} - \varepsilon) \geq \delta$.

Using the Bretagnolle–Huber inequality (see Theorem 14.2 of Lattimore and Szepesvári (2020)), we know

$$\frac{\mathbb{P}_1(E) + \mathbb{P}_2(E^c)}{2} \geq \frac{1}{4} \exp\left(-D_{KL}(\mathbb{P}_1, \mathbb{P}_2)\right).$$

By the chain rule for KL-divergence and the fact that $\mathbb{P}_1$ and $\mathbb{P}_2$ only differ in the reward for (s_{H-1}, a_{H-1}), we have

$$\begin{aligned}
&D_{KL}(\mathbb{P}_1, \mathbb{P}_2) \\
&= \mathbb{E}_1 \left[\sum_{i=1}^{n} \mathbb{I}\{S_{H-1}^{(i)} = s_{H-1}, A_{H-1}^{(i)} = a_{H-1}\} \left(\frac{1}{2} \log\left(\frac{1/2}{1/2 - \varepsilon}\right) + \frac{1}{2} \log\left(\frac{1/2}{1/2 + \varepsilon}\right) \right) \right] \\
&= \sum_{i=1}^{n} \mathbb{P}_1(S_{H-1}^{(i)} = s_{H-1}, A_{H-1}^{(i)} = a_{H-1}) \left(-\frac{1}{2} \log\left(1 - 4\varepsilon^2\right) \right) \\
&\leq \frac{n 8 \varepsilon^2}{A^H}
\end{aligned}$$

The last inequality follows from $-\log(1 - 4\varepsilon^2) \leq 8\varepsilon^2$ if $4\varepsilon^2 \leq 1/2$ (Krishnamurthy et al., 2016) and $\mathbb{P}_1(S_{H-1}^{(i)} = s_{H-1}, A_{H-1}^{(i)} = a_{H-1}) < 1/A^H$ from the construction of the MDPs.

Finally,

$$\frac{\mathbb{P}_1(E) + \mathbb{P}_2(E^c)}{2} \geq \frac{1}{4} \exp\left(-\frac{n 8 \varepsilon^2}{A^H}\right)$$

which is larger than δ if $n \leq A^H \ln(1/4\delta)/8\varepsilon^2$. As a result, we need at least $\Omega(A^H \ln(1/\delta)/\varepsilon^2)$ episodes. □

Corollary 2 (Upper bound on the sample complexity of OPS)**.** Suppose the data collection policy is uniformly random, that is, $\pi_b(a|s) = 1/A$ for all

$(s,a) \in \mathcal{S} \times \mathcal{A}$, and $|G_i| \leq V_{max}$ almost surely. Then the selection algorithm $\mathcal{L}$ that selects the policy with the highest IS estimate is (ε, δ)-sound with $O(A^H V_{max} \ln(|\Pi|/\delta)/\varepsilon^2)$ episodes.

Proof. Since the policy is uniform random, we know $|W_i G_i| < A^H V_{max}$ almost surely. Moreover, the importance sampling estimator is unbiased, that is, $\mathbb{E}[W_i G_i] = J(\pi)$. Using the Bernstein's inequality, we can show that the IS estimator satisfies

$$\Pr\left(\left|\hat{J}(\pi_k) - J(\pi_k)\right| \leq \frac{2A^H V_{max} \ln(2/\delta)}{3n} + \sqrt{\frac{2\mathrm{Var}(W_i G_i) \ln(2/\delta)}{n}}\right) \geq 1 - \delta$$

for one candidate policy π_k. Using the union bound over all candidate policies, we have

$$\Pr\left(\left|\hat{J}(\pi_k) - J(\pi_k)\right| \leq \frac{2A^H V_{max} \ln(2|\Pi|/\delta)}{3n} + \sqrt{\frac{2\mathrm{V}(W_i G_i) \ln(2|\Pi|/\delta)}{n}}, \forall k\right)$$
$$\geq 1 - \delta.$$

That is,

$$\Pr\left(J(\mathcal{L}(D,\Pi)) \geq J(\pi^\dagger) - \frac{4A^H V_{max} \ln(2|\Pi|/\delta)}{3n} + \sqrt{\frac{8\mathrm{V}(W_i G_i) \ln(2|\Pi|/\delta)}{n}}\right)$$
$$\geq 1 - \delta.$$

For the variance term,

$$\mathrm{V}(W_i G_i) = \mathbb{E}[W_i^2 G_i^2] - \mathbb{E}[W_i G_i]^2 \leq \mathbb{E}[W_i^2 G_i^2] \leq \sqrt{\mathbb{E}[W_i^2]\mathbb{E}[G_i^2]} \leq A^H V_{max}.$$

The second inequality follows from the Cauchy-Schwarz inequality. Therefore, if $n > 32A^H V_{max} \ln(2|\Pi|/\delta)/\varepsilon^2$, $\mathcal{L}$ is (ε, δ)-sound. □

A.1.4 Proof of Corollary 3

We first present the telescoping performance difference lemma, originally from Theorem 2 of Xie and Jiang (2020) for discounting setting, and Lemma 3.2 of Duan et al. (2021) for finite horizon setting.

Lemma A.2 (Lemma 3.2 of Duan et al. (2021))**.** Assume there exists a constant C such that, for any $\pi \in \Pi \cup \{\pi^*\}$, $\max_h \max_{s \in \mathcal{S}_h, a \in \mathcal{A}_h} \frac{d_h^\pi(s,a)}{\mu_h(s,a)} \leq C$. For

any $q \in \mathcal{Q}$, let π denotes the greedy policy with respect to $q = (q_0, \ldots, q_{H-1})$, then

$$J(\pi) \geq J(\pi^*) - 2H\sqrt{C}\sqrt{\frac{1}{H}\sum_{h=0}^{H-1} \|q_h - \mathcal{T} q_{h+1}\|_{2,\mu_h}} \ .$$

Corollary 3 (Error amplification for OPS using BE selection)**.** Suppose

1. there exists a constant C such that $\forall \pi \in \Pi \cup \{\pi^*\}$, $\max_{h\in[H]} \max_{s\in\mathcal{S}_h, a\in\mathcal{A}_h} \frac{d_h^\pi(s,a)}{\mu_h(s,a)} \leq C$,
2. the suboptimality of the candidate set is small, that is, $\min_{q\in\mathcal{Q}} \mathcal{E}(q) \leq \varepsilon_{sub}$,
3. there exists an $(\varepsilon_{est}, \delta)$-sound BE selection algorithm $\mathcal{L}$ on $\mathcal{Q}$,

then the OPS algorithm outputs the greedy policy with respect to $\mathcal{L}(D, \mathcal{Q})$ is $(2H\sqrt{C(\varepsilon_{sub} + \varepsilon_{est})}, \delta)$-sound.

Proof. Since $\mathcal{E}(q_k) = \frac{1}{H}\sum_h \|q_{k,h} - \mathcal{T} q_{k,h+1}\|_\mu^2 \leq \varepsilon_{sub} + \varepsilon_{est}$ with probability at least $1 - \delta$, Lemma A.2 implies that

$$J(\pi_k) \geq J(\pi^*) - 2H\sqrt{C(\varepsilon_{sub} + \varepsilon_{est})} \geq J(\pi^\dagger) - 2H\sqrt{C(\varepsilon_{sub} + \varepsilon_{est})}$$

where $\pi^\dagger$ is the best performing policy in Π.

By definition, the OPS algorithm is $(2H\sqrt{C(\varepsilon_{sub} + \varepsilon_{est})}, \delta)$-sound for this instance. □

A.1.5 Proof of Theorem 3

Assumption A.4 (Approximation error)**.** For any $h \in [H]$ and any candidate function $f \in \mathcal{Q}$ (here we use f instead of g to avoid confusion between q and g), we assume the approximation error is bounded by ε_{apx}, that is,

$$\inf_{g\in\mathcal{G}} \|g - \mathcal{T} f\|_{2,\mu_h}^2 \leq \varepsilon_{apx}.$$

Definition A.5 (Rademacher complexity)**.** Given a function class $\mathcal{F}$, let $X = \{x_1, \ldots, x_n\}$ denotes n fixed data points at horizon h following the distribution μ_h, the empirical Rademacher complexity is defined as

$$R_X(\mathcal{F}) = \mathbb{E}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i f(x_i) \mid X\right]$$

where the expectation is with respect to the Rademacher random variables σ_i. The Rademacher complexity is defined as $R_n^{\mu_h}(\mathcal{F}) = \mathbb{E}[R_X(\mathcal{F})]$ where the expectation is with respect to the n data points. Finally, to simply the notation, we define $R_n^{\mu}(\mathcal{F}) = \max_{h\in[H]} R_n^{\mu_h}(\mathcal{F})$ as the maximum Rademacher complexity over all horizons.

Definition A.6. Given an action value function f, define the state value function $v_f(s) = \arg\max_{a\in\mathcal{A}} f(s,a)$, and four loss functions:

$$\begin{aligned}
L_{D_h}(g,f) &:= \frac{1}{n}\sum_{(s,a,r,s')\in D_h} (g(s,a) - r - v_f(s'))^2 \\
L_D(g,f) &:= \frac{1}{H}\sum_h \frac{1}{n}\sum_{(s,a,r,s')\in D_h} (g(s,a) - r - v_f(s'))^2 \\
L_{\mu_h}(g,f) &:= \mathbb{E}_{(s,a,r,s')\sim\mu_h}[(g(s,a) - r - v_f(s'))^2] \\
L_{\mu}(g,f) &:= \frac{1}{H}\sum_h \mathbb{E}_{(s,a,r,s')\sim\mu}[(g(s,a) - r - v_f(s'))^2].
\end{aligned}$$

Theorem 9 (Bellman error selection in stochastic environments)**.** Suppose all $f \in \mathcal{Q}$ and $g \in \mathcal{G}$ take value in $[0, V_{max}]$. Let $k = \arg\min_i L_D(f_i, f_i) - L_D(\hat{g}_i, f_i)$ where $\hat{g}_i = \arg\min_{g\in\mathcal{G}} L_D(g, f_i)$. Then the following holds with probability at least $1-\delta$,

$$\mathcal{E}(q_k) \le \min_{i=1,\ldots,|\mathcal{Q}|} \mathcal{E}(q_i) + c_0\varepsilon_{apx} + c_0 V_{max}\mathcal{R}_n^{\mu}(\mathcal{G}) + c_0 V_{max}^2\sqrt{\frac{2\log(2|\Pi|H/\delta)}{n}}$$

for some constant $c_0 > 0$.

Proof. Fix a horizon h and a target function f. By concentration with Rademacher complexity (e.g., Lemma G.1 of Duan et al. (2021)), with probability at least $1-\delta/2$, we know, for any $g \in \mathcal{G}$,

$$|L_{D_h}(g,f) - L_{\mu_h}(g,f)| \le 2\mathcal{R}_n^{\mu_h}(L_{D_h}\circ\mathcal{G}) + V_{max}^2\sqrt{\frac{\log(2/\delta)}{2n}}.$$

We note that the loss function L_{D_h} is $(2V_{max})$-Lipschitz in its first argument, that is,

$$\begin{aligned}
&|L_{D_h}(g,f)(s,a,r,s') - L_{D_h}(g',f)(s,a,r,s')| \\
&= |(g(s,a) - r - V_f(s'))^2 - (g'(s,a) - r - V_f(s'))^2| \\
&= |(g(s,a) - g'(s,a))(g(s,a) + g'(s,a) - 2r - 2V_f(s'))| \\
&\leq |g(s,a) - g'(s,a)||g(s,a) + g'(s,a) - 2r - 2V_f(s')| \\
&\leq |g(s,a) - g'(s,a)|2V_{max}.
\end{aligned}$$

Therefore,

$$|L_{D_h}(g,f) - L_{\mu_h}(g,f)| \leq 4V_{max}\mathcal{R}_n^{\mu_h}(\mathcal{G}) + V_{max}^2\sqrt{\frac{\log(2/\delta)}{2n}}.$$

By Hoeffding's inequality, we also know, with probability at least $1-\delta/2$,

$$|L_{D_h}(f,f) - L_{\mu_h}(f,f)| \leq V_{max}^2\sqrt{\frac{\log(2/\delta)}{2n}}.$$

By the union bound over all $h \in [H]$, we have, with probability at least $1-\delta$,

$$\begin{aligned}
&|L_D(g,f) - L_\mu(g,f)| \leq 4V_{max}\mathcal{R}_n^{\mu}(\mathcal{G}) + V_{max}^2\sqrt{\frac{\log(2H/\delta)}{2n}}, \forall g \in \mathcal{G}, \text{ and} \\
&|L_D(f,f) - L_\mu(f,f)| \leq V_{max}^2\sqrt{\frac{\log(2H/\delta)}{2n}}.
\end{aligned}$$

Recall that we define $\hat{g}$ as the empirical minimizer, that is, $\hat{g} = \arg\min_{g\in\mathcal{G}} L_D(g,f)$, and let $g^\dagger$ be the population minimizer, that is, $g^\dagger = \arg\min_{g\in\mathcal{G}} L_\mu(g,f)$. It follows that with probability at least $1-\delta$,

$$\begin{aligned}
&|L_D(f,f) - L_D(\hat{g},f) - (L_\mu(f,f) - L_\mu(\mathcal{T}f,f))| \\
&\leq |L_D(f,f) - L_\mu(f,f)| + |L_D(\hat{g},f) - L_\mu(\mathcal{T}f,f)| \\
&\leq |L_D(f,f) - L_\mu(f,f)| + |\underbrace{L_D(\hat{g},f) - L_D(g^\dagger,f)}_{\leq 0}| + |L_D(g^\dagger,f) - L_\mu(g^\dagger,f)| + \\
&\quad \underbrace{|L_\mu(g^\dagger,f) - L_\mu(\mathcal{T}f,f)|}_{=\|g^\dagger - \mathcal{T}f\|_{2,\mu}^2} \\
&\leq |L_D(f,f) - L_\mu(f,f)| + |L_D(g^\dagger,f) - L_\mu(g^\dagger,f)| + \varepsilon_{apx} \\
&\leq \varepsilon_{apx} + 4V_{max}\mathcal{R}_n^{\mu}(\mathcal{G}) + V_{max}^2\sqrt{\frac{2\log(2H/\delta)}{n}}.
\end{aligned}$$

Note that $L_\mu(f,f) - L_\mu(\mathcal{T}f,f) = \mathcal{E}(f)$. Now apply the union bound for all f in the candidate set by setting $\delta = \delta/|\Pi|$, then the following holds for all f

$$|L_D(f,f) - L_D(\hat{g},f) - \mathcal{E}(f)| \leq \varepsilon_{apx} + 4V_{max}\mathcal{R}_n^\mu(\mathcal{G}) + V_{max}^2\sqrt{\frac{2\log(2|\Pi|H/\delta)}{n}}.$$

Let $k = \arg\min_i L_D(f_i,f_i) - L_D(\hat{g}_i,f_i)$, the following holds with probability at least $1-\delta$,

$$\mathcal{E}(f) \leq \min_{i=1,\ldots,|\mathcal{Q}|} \mathcal{E}(f_i) + c_0\varepsilon_{apx} + c_0 V_{max}\mathcal{R}_n^\mu(\mathcal{G}) + c_0 V_{max}^2\sqrt{\frac{2\log(2|\Pi|H/\delta)}{n}}$$

for some constant $c_0 > 0$.

Assume $\mathcal{F}$ has finite elements, and $\varepsilon_{apx} = 0$, then we need a sample size of $n = O(H^4 \log(|\mathcal{F}||\Pi|H/\delta)/\varepsilon_{est}^2)$. □

A.1.6 Sample complexity of FQE for OPS

Consider a function class $\mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_{H-1}$ such that $\mathcal{F}$ is closed under $\mathcal{T}^\pi$ for all $\pi \in \Pi$ and $|\mathcal{F}_h|$ is finite for all h. Assume $r_{max} = 1$ for simplicity. Given a policy $\pi \in \Pi$, we can show that

$$\begin{aligned}
&\|q_0^\pi - q_0\|_{1,d_0^\pi} \\
&= \sum_a \pi(a|s_0)|q_0^\pi(s_0,a) - q_0(s_0,a)| \\
&= \sum_a \pi(a|s_0)|(\mathcal{T}^\pi q_1^\pi)(s_0,a) - (\mathcal{T}^\pi q_1)(s_0,a) + (\mathcal{T}^\pi q_1)(s_0,a) - q_0(s_0,a)| \\
&\leq \sum_{a,s',a'} \pi(a|s_0)p(s'|s,a)\pi(a'|s')|q_1^\pi(s,a) - q_1(s,a)| + \sum_a \pi(a|s_0)|(\mathcal{T}^\pi q_1)(s_0,a) - q_0(s_0,a)| \\
&= \|q_1^\pi - q_1\|_{1,d_1^\pi} + \|\mathcal{T}^\pi q_1 - q_0\|_{1,d_0^\pi}.
\end{aligned}$$

Apply the same inequality recursively, we have

$$\begin{aligned}
\|q_0^\pi - q_0\|_{1,d_0^\pi} &\leq \sum_{h=0}^{H-1} \|\mathcal{T}^\pi q_{h+1} - q_h\|_{1,d_h^\pi} \\
&\leq H\sqrt{\frac{1}{H}\sum_{h=0}^{H-1} \|\mathcal{T}^\pi q_{h+1} - q_h\|_{2,d_h^\pi}^2} \\
&\leq H\sqrt{C\frac{1}{H}\sum_{h=0}^{H-1} \|\mathcal{T}^\pi q_{h+1} - q_h\|_{2,\mu_h}^2}\ .
\end{aligned}$$

The second inequality follows from the Cauchy-Schwarz inequality, and the last inequality follows from data coverage assumption.

Theorem 5.3 and Proposition 6.1 and of Duan et al. (2021) imply that for some constant c_0, with probability at least $1-\delta$,

$$\left|J(\pi)-\sum_{a\in\mathcal{A}}\pi(a|s_0)q_0(s_0,a)\right| \le c_0 H\sqrt{CH\left(\sum_h \frac{H\log|\mathcal{F}_h|}{n}+H^2\frac{\log(H/\delta)}{n}\right)}$$
$$\le c_0 H\sqrt{CH^2\frac{\log(|\mathcal{F}|H/\delta)}{n}}.$$

Apply the union bound, we know the following holds for all $\pi\in\Pi$ with probability at least $1-\delta$

$$\left|J(\pi)-\sum_{a\in\mathcal{A}}\pi(a|s_0)q_0(s_0,a)\right| \le c_0 H\sqrt{CH^2\frac{\log(|\mathcal{F}||\Pi|H/\delta)}{n}}.$$

To get an accuracy of $\varepsilon/2$, we need $n \ge c_1 H^4 C\log(|\mathcal{F}||\Pi|H/\delta)/\varepsilon^2$ for some constant c_1.

A.2 Experimental details

In this section, we provide the experimental details for classic RL environments and Atari environments.

A.2.1 FQE implementation

Since it is unclear how to perform model selection for FQE, we fix the function approximation as a two layer neural network model with hidden size 256 for classic RL experiment, and as a convolutional neural network followed by one layer neural network for atari experiments.

It is known that FQE can diverge, due to the fact that it combines off-policy learning with bootstrapping and function approximation, known as the deadly triad (Sutton & Barto, 2018). If one of the candidate policies is not well-covered, then the FQE estimate may overestimate the value of the uncovered policy (or even diverge to a very large value) and resulting in poor OPS. To circumvent the issue of uncovered policies, we need assign low value estimates for uncovered policies. In our FQE implementation, we assign low value

Algorithm 7 OPS using FQE

Input: Candidate set Π, training data D, function class $\mathcal{F}$, threshold U
for $\pi \in \Pi$ **do**
  Initialize $q_{H-1} = 0$
  for $h = H-2, \ldots, 0$, $q_h \leftarrow \arg\min_{f \in \mathcal{F}} \hat{l}_h(f, q_{h+1})$ where

$$\hat{l}_h(f, q_{h+1}) := \frac{1}{|D_h|} \sum_{(s,a,r,s') \in D_h} (f(s,a) - r - q_{h+1}(s', \pi(s')))^2$$

  Estimate the policy value $\hat{J}(\pi) \leftarrow \mathbb{E}_{a \sim \pi(\cdot|s_0)}[q_0(s_0, a)]$
  if $\hat{J}(\pi) > U$ **then**
    $\hat{J}(\pi) \leftarrow -\infty$
Output: $\pi^\dagger \leftarrow \arg\max_{\pi \in \Pi} \hat{J}(\pi)$

estimates to policies for which FQE diverges so the OPS algorithm would not choose these policies. We provide a pseudocode for OPS using FQE in Algorithm 7. In our experiment, we set $U = V_{max} + 100$.

A.2.2 Classic RL experiments

Stochastic environments. We implement stochastic environments by sticky actions. That is, when the agent selects an action to the environment, the action might be repeated with probability 25%, up to a maximum of 4 repeats.

Generating candidate policies. To generate a set of candidate policies, we run CQL with different hyperparameter configurations on a batch of data with 300 episodes collected with an ε-greedy policy with respect to a trained policy where $\varepsilon = 0.4$. The hyperparameter configuration includes:

- Learning rate $\in \{0.001, 0.0003, 0.0001\}$
- Network hidden layer size $\in \{128, 256, 512\}$
- Regularization coefficient $\in \{1.0, 0.1, 0.01, 0.001, 0.0\}$
- Iterations of CQL $\in \{100, 200\}$

As a result, we have 90 candidate policies.

Generating data for OPS. To generate data for offline policy selection, we use three different data distributions: (a) a data distribution collected by running the mixture of all candidate policies. As a result, the data distribution covers all candidate policies well; and (b) a data distribution collected by running the mixture of all candidate policies and an ε-greedy optimal policy that provides more diverse trajectories than (a).

Randomness across multiple runs. To perform experiments with multiple runs, we fix the offline data and the candidate policies and only resample the offline data for OPS. This better reflects the theoretical result that the randomness is from resampling the data for an OPS algorithm. In our experiments, we use 10 runs and report the average regret with one standard error. Since the variability across runs is not large, we find using 10 runs is enough.

Random selection baseline. We include a random selection baseline that randomly chooses k policies. Since the random selection algorithm has very high variance, we estimate the expected regret of random selection by repeating the random selection 10000 times, and report the average regret.

BVFT. BVFT has a hyperparameter: the discretization resolution. We follow the method described in the original paper to search for the best resolution from a set of predefined values. Note that in the authors' implementation, they use different sets for different environments.

A.2.3 Atari experiments

Generating candidate policies. To generate a set of candidate policies, we run CQL with the hyperparameters used in the original paper:

- Regularization coefficient $\in \{0.5, 4.0, 5.0\}$
- Number of gradient steps $\in \{50k, 100k, 150k, \ldots, 1000k\}$

Randomness across multiple runs. Similar to the classic RL experiments, we fix the candidate policies and only resample the offline data for OPS. In our experiments, we use 5 different DQN replay dataset and report the average regret with one standard error.

Appendix B

Supplementary Material for FQI-AIR

We provide the proof for the theoretical results presented in Chapter 5.

B.1 Theoretical analysis

We first provide lemmas that would be useful for proving our main theorems.

Lemma 1. Given a MDP M, suppose the sequence of value function $(q_0, \ldots, q_{H-1})$ satisfies that $\|\mathcal{T}q_{h+1} - q_h\|_{2,d_h^\pi} \leq \varepsilon$ for all policy π with $q_H = 0$. Let π be the greedy policy with respect to $(q_0, \ldots, q_{H-1})$, then we have that

$$J(\pi, M) \geq J(\pi_M^*, M) - (H+1)H\varepsilon.$$

Proof. By the performance difference lemma (Lemma 5.2.1 of Kakade (2003)

or Chen and Jiang (2019)), we get

$$
\begin{aligned}
&J(\pi^*, M_D) - J(\pi, M_D) \\
&= \mathbb{E}^{\pi}\left[\sum_{h=0}^{H-1} q^*(S_h, \pi^*(S_h)) - q^*(S_h, \pi_h(S_h))\right] \\
&\leq \mathbb{E}^{\pi}\left[\sum_{h=0}^{H-1} q^*(S_h, \pi^*(S_h)) - q_h(S_h, \pi^*(S_h)) + q_h(S_h, \pi_h(S_h)) - q^*(S_h, \pi_h(S_h))\right] \\
&\leq \mathbb{E}^{\pi}\left[\sum_{h=0}^{H-1} |q^*(S_h, \pi^*(S_h)) - q_h(S_h, \pi^*(S_h))| + |q_h(S_h, \pi_h(S_h)) - q^*(S_h, \pi_h(S_h))|\right] \\
&\leq \sum_{h=0}^{H-1} \|q^* - q_h\|_{1, d_h^{\pi}\pi^*} + \|q^* - q_h\|_{1, d_h^{\pi}\pi_h} \\
&\leq \sum_{h=0}^{H-1} \|q^* - q_h\|_{2, d_h^{\pi}\pi^*} + \|q^* - q_h\|_{2, d_h^{\pi}\pi_h}
\end{aligned}
$$

where d_h^{π} is the state-action distribution at horizon h induced by policy π. The first inequality follows because π_h is greedy with respect to q_h, so for any $s_h \in \mathcal{S}_h$, $q_h(s_h, \pi^*(s_h)) \leq q_h(s_h, \pi_h(s_h))$. The last inequality follows from the Jensen's inequality.

Consider a state-action distribution β_0 that is induced by some policy, then

$$
\begin{aligned}
\|q^* - q_0\|_{2,\beta_0} &\leq \|\mathcal{T}q^* - \mathcal{T}q_1 + \mathcal{T}q_1 - q_0\|_{2,\beta_0} \\
&\leq \|\mathcal{T}q^* - \mathcal{T}q_1\|_{2,\beta_0} + \|\mathcal{T}q_1 - q_0\|_{2,\beta_0} \\
&\leq \|q^* - q_1\|_{2,\beta_1} + \varepsilon
\end{aligned}
$$

where

$$
\beta_1(s', a') = \sum_{s,a} \beta_0(s, a) P(s, a, s') \mathbb{I}\{a' = \arg\max_{a'' \in \mathcal{A}} (q^*(s', a'') - q_1(s', a''))^2\}
$$

is also induced by some policy. The first inequality follows by the fact that q^* is the fixed point of the operator $\mathcal{T}$. We can recursively apply the same process for $\|q^* - q_h\|_{2,\beta_h}$, $h > 0$, and we can get

$$
\|q^* - q_h\|_{2,\beta_h} \leq (H - h)\varepsilon.
$$

Plug in the inequality to the performance difference lemma, we get

$$
J(\pi^*, M_D) - J(\pi, M_D) \leq \sum_{h=0}^{H-1} 2(H - h)\varepsilon = (H + 1)H\varepsilon.
$$

□

The simulation lemma was first introduced for discounted setting in Kearns and Singh (2002), and here we prove a modified version of the simulation lemma for the finite horizon MDPs.

Lemma 2 (Finite Horizon Simulation Lemma)**.** Given a policy π, let $M = (\mathcal{S}, \mathcal{A}, P, r, H, \mu)$ and $\hat{M} = (\mathcal{S}, \mathcal{A}, \hat{P}, r, H, \mu)$ be two finite horizon MDPs and π be a policy satisfying that for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$, $\|P(s,a) - \hat{P}(s,a)\|_1 \leq \varepsilon$. Then, $|v_M^\pi(s) - v_{\hat{M}}^\pi(s)| \leq \varepsilon \frac{H^2 r_{max}}{2}$ for $s \in \mathcal{S}_0$.

Proof. For all $s \in \mathcal{S}_0, a \in \mathcal{A}$,

$$
\begin{aligned}
&|v_M^\pi(s) - v_{\hat{M}}^\pi(s)| \\
&= |r_\pi(s) + \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}_1} P(s,a,s') v_M^\pi(s') - r_\pi(s) - \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}_1} \hat{P}(s,a,s') v_{\hat{M}}^\pi(s')| \\
&= |\sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}_1} P(s,a,s') v_M^\pi(s') - \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}_1} \hat{P}(s,a,s') v_{\hat{M}}^\pi(s')| \\
&= |\sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}_1} P(s,a,s') v_M^\pi(s') - \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}_1} \hat{P}(s,a,s') v_M^\pi(s') \\
&\quad + \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}_1} \hat{P}(s,a,s') v_M^\pi(s') - \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}_1} \hat{P}(s,a,s') v_{\hat{M}}^\pi(s')| \\
&\leq \sum_{a\in\mathcal{A}} \pi(a|s) |\sum_{s'\in\mathcal{S}_1} (P(s,a,s') - \hat{P}(s,a,s')) v_M^\pi(s')| + \max_{s'\in\mathcal{S}_1} |v_M^\pi(s') - v_{\hat{M}}^\pi(s')| \\
&\leq \varepsilon(H-1) r_{max} + \max_{s'\in\mathcal{S}_1} |v_M^\pi(s') - v_{\hat{M}}^\pi(s')| \\
&\leq \varepsilon(H-1) r_{max} + \varepsilon_P (H-2) r_{max} + \cdots + \varepsilon_P 1 r_{max} \\
&\leq \varepsilon \frac{H^2 r_{max}}{2} = \varepsilon \frac{H v_{max}}{2}
\end{aligned}
$$

The first equality follows from the Bellman equation. The second and third inequalities follow from that $v_{\hat{M}}^\pi(s)$ is at most $(H-h) r_{max}$ for $s \in \mathcal{S}_h$. □

Lemma 3. Let $M = (\mathcal{S}, \mathcal{A}, P^{\text{exo}}, P^{\text{end}}, r, H)$ be an ε_{air}-AIR MDP and $M_b = (\mathcal{S}, \mathcal{A}, \tilde{P}^{\text{exo}}, \tilde{P}^{\text{end}}, r, H)$ with $D_{TV}(P^{\text{end}}(s,a), \tilde{P}^{\text{end}}(s,a)) \leq \varepsilon_p$, then for any policy π,

$$|J(\pi, M) - J(\pi, M_b)| \leq v_{max} H (\varepsilon_{air} + \varepsilon_p).$$

Proof. Since M is ε_{air}-AIR, we have

$$D_{TV}\left(P^{\text{exo}}(s^{\text{exo}},a),P^{\text{exo}}(s^{\text{exo}},a')\right)=\frac{1}{2}\left\|P^{\text{exo}}(s^{\text{exo}},a)-P^{\text{exo}}(s^{\text{exo}},a')\right\|_1\leq\varepsilon_{air}$$

Let $e(s,a,s'^{\text{end}})=P^{\text{end}}(s,a,s'^{\text{end}})-\tilde{P}^{\text{end}}(s,a,s'^{\text{end}})$, we know

$$D_{TV}\left(P^{\text{end}}(s,a),\tilde{P}^{\text{end}}(s,a)\right)=\frac{1}{2}\left\|e(s,a,\cdot)\right\|_1=\frac{1}{2}\sum_{s'^{\text{end}}}|e(s,a,s'^{\text{end}})|\leq\varepsilon_p.$$

For any $s=(s^{\text{exo}},s^{\text{end}})\in\mathcal{S}^{\text{exo}}\times\mathcal{S}^{\text{end}}$ and $a\in\mathcal{A}$, we have

$$\begin{aligned}
&\left\|P(s,a)-\tilde{P}(s,a)\right\|_1\\
&=\sum_{s'}|P([s^{\text{exo}},s^{\text{end}}],a,s')-\tilde{P}([s^{\text{exo}},s^{\text{end}}],a,s')|\\
&=\sum_{s'}|P^{\text{exo}}(s^{\text{exo}},a,s'^{\text{exo}})P^{\text{end}}(s,a,s'^{\text{end}})-\tilde{P}^{\text{exo}}(s^{\text{exo}},a,s'^{\text{exo}})\tilde{P}^{\text{end}}(s,a,s'^{\text{end}})|\\
&=\sum_{s'}|P^{\text{exo}}(s^{\text{exo}},a,s'^{\text{exo}})(\tilde{P}^{\text{end}}(s,a,s'^{\text{end}})+e(s,a,s'^{\text{end}}))\\
&\quad-\tilde{P}^{\text{exo}}(s^{\text{exo}},a,s'^{\text{exo}})\tilde{P}^{\text{end}}(s,a,s'^{\text{end}})|\\
&=\sum_{s'}|\tilde{P}^{\text{end}}(s,a,s'^{\text{end}})[P^{\text{exo}}(s^{\text{exo}},a,s'^{\text{exo}})-\tilde{P}^{\text{exo}}(s^{\text{exo}},a,s'^{\text{exo}})]\\
&\quad+P^{\text{exo}}(s^{\text{exo}},a,s'^{\text{exo}})e(s,a,s'^{\text{end}})|\\
&\leq\sum_{s'}\tilde{P}^{\text{end}}(s,a,s'^{\text{end}})|P^{\text{exo}}(s^{\text{exo}},a,s'^{\text{exo}})-\sum_{a'}\pi'(a'|s^{\text{exo}})P^{\text{exo}}(s^{\text{exo}},a',s'^{\text{exo}})|\\
&\quad+\sum_{s'}P^{\text{exo}}(s^{\text{exo}},a,s'^{\text{exo}})|e(s,a,s'^{\text{end}})|\\
&\leq\sum_{a'}\pi'(a|s^{\text{exo}})\sum_{s'^{\text{exo}}}|P^{\text{exo}}(s^{\text{exo}},a,s'^{\text{exo}})-P^{\text{exo}}(s^{\text{exo}},a',s'^{\text{exo}})|\\
&\quad+\sum_{s'^{\text{end}}}|e(s,a,s'^{\text{end}})|\\
&\leq 2\varepsilon_{air}+2\varepsilon_p.
\end{aligned}$$

The first inequality follows by writing

$$\tilde{P}^{\text{exo}}(s^{\text{exo}},a,s'^{\text{exo}})=\sum_{a'\in\mathcal{A}}\pi'(a'|s^{\text{exo}})P^{\text{exo}}(s^{\text{exo}},a',s'^{\text{exo}})$$

where $\pi'(a'|s_h^{\text{exo}})=\frac{\mathbb{P}_M^{\pi_b}(S_h^{\text{exo}}=s_h^{\text{exo}},A_h=a')}{\mathbb{P}_M^{\pi_b}(S_h^{\text{exo}}=s_h^{\text{exo}})}$ when the denominator is nonzero, and otherwise let $\pi'(\cdot|s_h^{\text{exo}})$ be an arbitrary distribution. Applying Lemma 2, we get

$$|J(\pi,M)-J(\pi,M_b)|=\sum_{s_0}\mu(s_0)|v_M^{\pi}(s_0)-v_{M_b}^{\pi}(s_0)|\leq v_{max}(\varepsilon_{air}+\varepsilon_p)H.$$

□

Theorem 4 (Performance bound for FQI-AIR)**.** Under Assumption 1 and 2, let π_M^* be an optimal policy in M and π the output policy of FQI-AIR, then with probability at least $1-\zeta$

$$J(\pi, M) \geq J(\pi_M^*, M) - 2v_{max}H(\varepsilon_{air} + \varepsilon_p) - \\ (H+1)H\sqrt{|\mathcal{S}^{\text{end}}||\mathcal{A}|}\left(\sqrt{\frac{72{v_{max}}^2 \ln\left(H|\mathcal{F}||\mathcal{S}^{\text{end}}||\mathcal{A}|/\zeta\right)}{N}} + 2\varepsilon_{apx}\right).$$

Proof. First fix a horizon $h \in [H]$ and $f' = q_{h+1} \in \mathcal{F}$. Define

$$R(f) \\ = \|\mathcal{T}f' - f\|_{2,\tilde{\nu}_h}^2 \\ = \sum_{s_h^{\text{exo}} \in S_h^{\text{exo}}} \sum_{s_h^{\text{end}} \in S_h^{\text{end}}} \sum_{a \in \mathcal{A}} \frac{\tilde{\nu}_h(s_h^{\text{exo}})}{|\mathcal{S}^{\text{end}}||\mathcal{A}|}(f(s_h^{\text{exo}}, s_h^{\text{end}}, a) - r - \mathbb{E}[\max_{a'} f'(S_{h+1}^{\text{exo}}, S_{h+1}^{\text{end}}, a')])^2$$

and

$$R_n(f) \\ = \frac{1}{N}\sum_{i=1}^{N} \sum_{s_h^{\text{end}} \in S_h^{\text{end}}} \sum_{a \in \mathcal{A}} \frac{1}{|\mathcal{S}^{\text{end}}||\mathcal{A}|}(f(s_h^{(i),\text{exo}}, s_h^{\text{end}}, a) - r - \max_{a'} f'(s_{h+1}^{(i),\text{exo}}, s_{h+1}^{\text{end}}, a'))^2$$

where $s_{h+1}^{\text{end}} \sim \hat{P}^{\text{end}}(s_h^{(i),\text{exo}}, s_h^{\text{end}}, a)$.

Let

$$\hat{f} = \arg\min_{f \in \mathcal{F}} R_n(f), \tilde{f} = \arg\min_{f \in \mathcal{F}} R(f),$$

our goal is to bound $R(\hat{f})$ with high probability. This is similar to bounding the generalization error in the statistical learning literature. We follow the proof technique of Lemma A.11 in A. Agarwal et al. (2019) to bound the excess risk.

Fix $u = (s_h^{\text{end}}, a) \in \mathcal{U} := \mathcal{S}^{\text{end}} \times \mathcal{A}$, let $x_i^u = (s_h^{\text{exo}}, s_h^{\text{end}}, a)$ and $y_i^u = r(s_h^{\text{exo}}, s_h^{\text{end}}, a) + \max_{a'} f'(s_{h+1}^{\text{exo}}, s_{h+1}^{\text{end}}, a')$ with $(x_i^u, y_i^u) \sim \nu$, $f^*(x_i^u) = \mathbb{E}[y_i^u | x_i^u]$. Note that our goal is to bound $R(\hat{f}) = \frac{1}{|\mathcal{U}|}\sum_{u \in \mathcal{U}} \mathbb{E}[(\hat{f}(x_i^u) - f^*(x_i^u))^2]$.

First note that

$$\mathbb{E}[(f(x_i^u) - y_i^u)^2 - (f^*(x_i^u) - y_i^u)^2] = \mathbb{E}[(f(x_i^u) - f^*(x_i^u))^2]$$

and

$$\begin{aligned}\mathrm{V}^{u,f} &:= \mathrm{V}[(f(x_i^u) - y_i^u)^2 - (f^*(x_i^u) - y_i^u)^2] \\ &\leq \mathbb{E}[((f(x_i^u) - y_i^u)^2 - (f^*(x_i^u) - y_i^u)^2)^2] \\ &\leq 4{v_{max}}^2 \mathbb{E}[(f(x_i^u) - f^*(x_i^u))^2].\end{aligned}$$

We can bound the deviation from the mean for all $f \in \mathcal{F}$ with one-sided Bernstein's inequality: the following holds with probability $1 - \zeta$,

$$\begin{aligned}&\mathbb{E}[(f(x_i^u) - f^*(x_i^u))^2] - \frac{1}{N}\sum_{i=1}^{N}[(f(x_i^u) - y_i^u)^2 - (f^*(x_i^u) - y_i^u)^2] \\ &\leq \sqrt{\frac{2\mathrm{V}^{u,f}\ln(|\mathcal{F}|/\zeta)}{N}} + \frac{4{v_{max}}^2\ln(|\mathcal{F}|/\zeta)}{3N} \\ &\leq \sqrt{\frac{8{v_{max}}^2\mathbb{E}[(f(x_i^u) - f^*(x_i^u))^2]\ln(|\mathcal{F}|/\zeta)}{N}} + \frac{4{v_{max}}^2\ln(|\mathcal{F}|/\zeta)}{3N}.\end{aligned}$$

We need this holds for all $u \in \mathcal{U}$. By the union bound, we have for all $u \in \mathcal{U}$

$$\begin{aligned}&\mathbb{E}[(f(x_i^u) - f^*(x_i^u))^2] - \frac{1}{N}\sum_{i=1}^{N}[(f(x_i^u) - y_i^u)^2 - (f^*(x_i^u) - y_i^u)^2] \\ &\leq \sqrt{\frac{8{v_{max}}^2\mathbb{E}[(f(x_i^u) - f^*(x_i^u))^2]\ln(|\mathcal{F}||\mathcal{U}|/\zeta)}{N}} + \frac{4{v_{max}}^2\ln(|\mathcal{F}||\mathcal{U}|/\zeta)}{3N}.\end{aligned}$$

Since $\hat{f}$ is the empirical minimizer, we have

$$\begin{aligned}&\frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{1}{N}\sum_{i=1}^{N}[(\hat{f}(x_i^u) - y_i^u)^2 - (f^*(x_i^u) - y_i^u)^2] \\ &\leq \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{1}{N}\sum_{i=1}^{N}[(\tilde{f}(x_i^u) - y_i^u)^2 - (f^*(x_i^u) - y_i^u)^2].\end{aligned}$$

Since $\tilde{f} \in \mathcal{F}$, we have

$$\begin{aligned}&\frac{1}{N}\sum_{i=1}^{N}[(\tilde{f}(x_i^u) - y_i^u)^2 - (f^*(x_i^u) - y_i^u)^2] - \mathbb{E}[(\tilde{f}(x_i^u) - f^*(x_i^u))^2] \\ &\leq \sqrt{\frac{8{v_{max}}^2\mathbb{E}[(\tilde{f}(x_i^u) - f^*(x_i^u))^2]\ln(|\mathcal{F}||\mathcal{U}|/\zeta)}{N}} + \frac{4{v_{max}}^2\ln(|\mathcal{F}||\mathcal{U}|/\zeta)}{3N}.\end{aligned}$$

Since $\frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\mathbb{E}[(\tilde{f}(x_i^u)-f^*(x_i^u))^2]\leq\varepsilon_{apx}$, we have

$$\begin{aligned}
&\frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{1}{N}\sum_{i=1}^{N}[(\tilde{f}(x_i^u)-y_i^u)^2-(f^*(x_i^u)-y_i^u)^2]\\
&\leq\varepsilon_{apx}+\sqrt{\frac{8{v_{max}}^2\varepsilon_{apx}\ln(|\mathcal{F}||\mathcal{U}|/\zeta)}{N}}+\frac{4{v_{max}}^2\ln(|\mathcal{F}||\mathcal{U}|/\zeta)}{3N}\\
&\leq\frac{3}{2}\varepsilon_{apx}+\frac{16{v_{max}}^2\ln(|\mathcal{F}||\mathcal{U}|/\zeta)}{3N}.
\end{aligned}$$

The second inequality follows by the fact that $\sqrt{ab}\leq\frac{a+b}{2}$ for $a,b\geq 0$. Then,

$$\begin{aligned}
&\frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\mathbb{E}[(\hat{f}(x_i^u)-f^*(x_i^u))^2]\\
&\leq\sqrt{\frac{8{v_{max}}^2\frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\mathbb{E}[(\hat{f}(x_i)-f^*(x_i))^2]\ln(|\mathcal{F}||\mathcal{U}|/\zeta)}{N}}+\frac{20{v_{max}}^2\ln(|\mathcal{F}||\mathcal{U}|/\zeta)}{3N}\\
&\quad+\frac{3}{2}\varepsilon_{apx}.
\end{aligned}$$

Solving for the quadratic formula, we get

$$\frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\mathbb{E}[(\hat{f}(x_i^u)-f^*(x_i^u))^2]\leq\frac{36{v_{max}}^2\ln(|\mathcal{F}||\mathcal{U}|/\zeta)}{N}+2\varepsilon_{apx}.$$

Now define $\tilde{d}_h^\pi(s^{\text{exo}})=\mathbb{P}_{M_b}^\pi(S_h^{\text{exo}}=s_h^{\text{exo}})$ and $d_h^\pi(s^{\text{exo}},s^{\text{end}},a)=\mathbb{P}_{M_b}^\pi(S_h^{\text{exo}}=s_h^{\text{exo}},S_h^{\text{end}}=s_h^{\text{exo}},A_h=a)$. By the construction of M_b, we have

$$\begin{aligned}
&\forall\pi,s_h^{\text{exo}}\in S_h^{\text{exo}},\frac{\tilde{d}_h^\pi(s^{\text{exo}})}{\tilde{\nu}_h(s^{\text{exo}})}\leq 1,\text{ and}\\
&\forall\pi,s_h^{\text{exo}}\in S_h^{\text{exo}},s_h^{\text{end}}\in S_h^{\text{end}},a\in\mathcal{A},\frac{d_h^\pi(s^{\text{exo}},s^{\text{end}},a)}{\tilde{\nu}_h(s^{\text{exo}})/|\mathcal{S}^{\text{end}}||\mathcal{A}|}\leq|\mathcal{S}^{\text{end}}||\mathcal{A}|.
\end{aligned}$$

Therefore, for any policy π, we have with probability $1-\zeta$

$$\begin{aligned}
\|\mathcal{T}q_{h+1}-q_h\|_{2,d_h^\pi}&\leq\sqrt{|\mathcal{S}^{\text{end}}||\mathcal{A}|}\|\mathcal{T}q_{h+1}-q_h\|_{2,\tilde{\nu}_h}\\
&\leq\sqrt{|\mathcal{S}^{\text{end}}||\mathcal{A}|}(\sqrt{\frac{36{v_{max}}^2\ln(|\mathcal{F}||\mathcal{S}^{\text{end}}||\mathcal{A}|/\zeta)}{N}+2\varepsilon_{apx}}).
\end{aligned}$$

Note that this holds for a fixed $h\in[H]$ and $f'\in\mathcal{F}$. Use the union bound, we have that, for any policy π, $h\in[H]$ and $q_{h+1}\in\mathcal{F}$, with probability $1-\zeta$

$$\|\mathcal{T}q_{h+1}-q_h\|_{2,d_h^\pi}\leq\sqrt{|\mathcal{S}^{\text{end}}||\mathcal{A}|}(\sqrt{\frac{72{v_{max}}^2\ln(H|\mathcal{F}||\mathcal{S}^{\text{end}}||\mathcal{A}|/\zeta)}{N}+2\varepsilon_{apx}}).$$

By Lemma 1, the output policy π satisfies

$$
\begin{aligned}
&J(\pi^*_{M_b}, M_b) - J(\pi, M_b) \\
&\leq (H+1)H\sqrt{|\mathcal{S}^{\text{end}}||\mathcal{A}|}\left(\sqrt{\frac{72{v_{max}}^2 \ln\left(H|\mathcal{F}||\mathcal{S}^{\text{end}}||\mathcal{A}|/\zeta\right)}{N}} + 2\varepsilon_{apx}\right) = \varepsilon_1.
\end{aligned}
$$

By Lemma 3, the output policy π satisfies

$$
\begin{aligned}
&J(\pi^*_M, M) - J(\pi, M) \\
&= J(\pi^*_M, M) - J(\pi^*_M, M_b) + J(\pi^*_M, M_b) - J(\pi, M) \\
&\leq J(\pi^*_M, M) - J(\pi^*_M, M_b) + J(\pi^*_{M_b}, M_b) - J(\pi, M) \\
&\leq J(\pi^*_M, M) - J(\pi^*_M, M_b) + J(\pi, M_b) - J(\pi, M) + \varepsilon_1 \\
&\leq |J(\pi^*_M, M) - J(\pi^*_M, M_b)| + |J(\pi, M_b) - J(\pi, M)| + \varepsilon_1 \\
&\leq 2v_{max}H(\varepsilon_{air} + \varepsilon_P) \\
&\quad + (H+1)H\sqrt{|\mathcal{S}^{\text{end}}||\mathcal{A}|}\left(\sqrt{\frac{72{v_{max}}^2 \ln\left(H|\mathcal{F}||\mathcal{S}^{\text{end}}||\mathcal{A}|/\zeta\right)}{N}} + 2\varepsilon_{apx}\right).
\end{aligned}
$$

Note that $\sqrt{|\mathcal{S}^{\text{end}}||\mathcal{A}|}$ comes from the fact that we run FQI for all endogenous state and action uniformly. This term can be viewed as the concentration coefficient in the standard FQI analysis and is unavoidable. However, if we can adapt the weight on the endogenous state and action, we might be able to reduce the dependence. □

Theorem 5. Under Assumption 1, given a deterministic policy π, we have that with probability at least $1-\zeta$

$$
\left|\hat{J}(\pi, M) - J(\pi, M)\right| \leq v_{max}\left(H\varepsilon_{air} + H\varepsilon_p + \sqrt{\frac{\ln(2/\zeta)}{2N}}\right).
$$

Proof. We first show that $R(\tau_D^{(i)}) := \sum_{t=0}^{H-1} r(s_t^{(i)}, a_t^{(i)})$ for $i \in [N]$ are i.i.d. samples with mean $J(\pi, M_b)$. Let $\mathbb{P}_D^\pi$ be the probability measure on trajectories τ_D. Note that the randomness of τ_D comes from the generation of exogenous variables in by the interaction between π_b and M, and the generation of actions and endogenous variables by π and $\hat{P}^{\text{end}}$. Let $\mathbb{P}_{M_b}^\pi$ be the probability measure on the trajectories sampled by running π on M_b. It is sufficient to show that $\mathbb{P}_D^\pi = \mathbb{P}_{M_b}^\pi$ then $\mathbb{E}[R(\tau_D)] = \int R(\tau) d\mathbb{P}_D^\pi(\tau) = \int R(\tau) d\mathbb{P}_{M_b}^\pi(\tau) = J(\pi, M_b)$.

For all trajectories $\tau = (s_0, a_0, \ldots, s_{H-1}, a_{H-1}) \in (\mathcal{S} \times \mathcal{A})^H$, we have

$$
\begin{aligned}
&\mathbb{P}_D^{\pi}(\tau) \\
&= \mathbb{P}_M^{\pi_b}(s_0, s_1^{\text{exo}}, \ldots, s_{H-1}^{\text{exo}}) \mathbb{P}_D^{\pi}(a_0, s_1^{\text{end}}, \ldots, s_{H-1}^{\text{end}}, a_{H-1} \mid s_0, s_1^{\text{exo}}, \ldots, s_{H-1}^{\text{exo}}) \\
&= \mu(s_0)\pi(a_0|s_0)\mathbb{P}_M^{\pi_b}(s_1^{\text{exo}}|s_0)\hat{P}^{\text{end}}(s_0, a_0, s_1^{\text{end}})\pi(a_1|s_1)\ldots \\
&= \mu(s_0)\pi(a_0|s_0)\tilde{P}(s_0, a_0, s_1)\pi(a_1|s_1)\cdots = \mathbb{P}_{M_b}^{\pi}(\tau).
\end{aligned}
$$

By Hoeffding's inequality, with probability at least $1 - \zeta$,

$$
\left|\hat{J}(\pi, M) - J(\pi, M_b)\right| = \left|\frac{1}{N}\sum_{i=1}^{N} R(\tau_D^{(i)}) - \mathbb{E}[R(\tau_D)]\right| \leq v_{max}\sqrt{\frac{1}{2N}\ln\frac{2}{\zeta}}.
$$

Finally by Lemma 3, the followings hold with probability at least $1 - \zeta$:

$$
\begin{aligned}
|\hat{J}(\pi, M) - J(\pi, M)| &\leq |\hat{J}(\pi, M) - J(\pi, M_b)| + |J(\pi, M_b) - J(\pi, M)| \\
&\leq v_{max}H(\epsilon_{AIR} + \epsilon_P) + v_{max}\sqrt{\frac{\ln(2/\zeta)}{2N}}.
\end{aligned}
$$

□

Appendix C

Supplementary Material for Regression-Assisted Estimators

We provide supplementary material for Chapter 6.

C.1 Overview of survey sampling

In this section, we introduce the survey sampling terminology and how to use it for OPE. The terminology will be used for proving our theoretical results.

Survey sampling can be dated back to Hansen and Hurwitz, 1943; Horvitz and Thompson, 1952, where they consider the problem of selecting a sample of units from a finite population to estimate unknown population parameters. For example, if the goal is to estimate the customer satisfaction rate for a product, survey sampling is concerned with selecting a subset of customers to conduct surveys. Since then, the field has investigated a variety of practical scenarios, including dealing with missing data, handling nonstationarity and understanding to how to leverage auxiliary information.

Formally, let $\mathcal{U} = \{1, \ldots, N\}$ be the population of interest, y_i be the study variable and $\mathbf{x}_i$ be the auxiliary variable for the unit $i \in \mathcal{U}$. Continuing the above example, the population could be all customers, the study variable could be the satisfaction level, and the auxiliary variable could be the information about the customer. A subset of the population, called a sample, is selected according to a sampling design. We observe the study variable for units in the sample, and the goal is to estimate the population total of the study variables

$t_y = \sum_{i \in \mathcal{U}} y_i$.

A sampling design $\mathbf{I} = (I_1, \ldots, I_N)$ is a random vector describing how the sample is drawn from the population: $I_i > 0$ means that the unit i is selected in the sample and $I_i = 0$ means the unit is not selected. For example, a multinomial design is a with-replacement and fixed-size design where we draw n units independently and identically according to probability p_i with $\sum_{i \in \mathcal{U}} p_i = 1$. In this case, the design vector $\mathbf{I}$ follows the multinomial distribution with parameters n and $(p_i)_{i=1}^N$, that is, $P(\mathbf{I}_1 = i_i, \ldots, \mathbf{I}_N = i_N) = \frac{n!}{\Pi_{i=1}^N i_i!} p_i^{i_i} \ldots p_N^{i_N}$ if $\sum_i i_i = n$ and 0 otherwise. In survey sampling, the study variable is fixed and the randomness comes from the sampling design $\mathbf{I}$.

Given a sample D of fixed size n, the Hansen-Hurwitz (HH) estimator (Hansen & Hurwitz, 1943) for multinomial design is $\hat{t}_{\text{HH}} = \sum_{i \in D} \frac{y_i}{\mathbb{E}[I_i]} = \sum_{i \in D} \frac{y_i}{np_i}$. The estimator $\hat{t}_{\text{HH}}$ is an unbiased estimator for t_y if $p_i > 0$ for all $i \in \mathcal{U}$.

This formalize OPE under survey sampling, let the population be $\mathcal{U} = \mathcal{S} \times \mathcal{A}$ and the study variable be $y_{s,a} = P(s)\pi(a|s)r(s,a)$. The population total of y is the value of the policy: $t_y = \sum_{(s,a) \in \mathcal{U}} y_{s,a} = J(\pi)$. The weighting $P(s)\pi(a|s)$ goes into the study variable since the goal is to estimate the total of study variable without weighting. Even though we have $P(s)$ in the study variable, the term often cancels out as we will see for the HH estimator.

For OPE, the sampling design is the multinomial design with sampling probability $p_{s,a} = P(s)\pi(a|s)$. Given a sample $D = \{(s_i, a_i, r(s_i, a_i))\}_{i=1}^n$, the HH estimator is

$$\hat{t}_{\text{HH}} = \sum_{(s,a) \in D} \frac{y_{s,a}}{np_{s,a}} = \sum_{(s,a) \in D} \frac{P(s)\pi(a|s)r(s,a)}{nP(s)\pi_b(a|s)} = \frac{1}{n} \sum_{(s,a) \in D} \frac{\pi(a|s)}{\pi_b(a|s)} r(s,a).$$

It has the same form as the IS estimator. In the case where the sampling design p is not known and needs to be estimated by a propensity model, it is called the inverse propensity score (IPS) estimator.

The HH estimator is called the *design-based* estimator in the survey sampling literature. This is because the primary source of randomness is from the sampling design. Another approach is called the *model-based* approach which assumes the study variables are generated by a superpopulation model. The

goal is to model the relationship between the study variable and the auxiliary variable. The resulting estimator is similar to the direct method.

The model-based approach. The model-based approach is a popular approach in the survey sampling literature. Chambers and Clark, 2012 provide an introduction for the model-based approach. Different from design-based and model-assisted approaches, the study variables are assumed to be generated by a superpopulation model and typically depends on the auxiliary variable. More previously, we assume the values $y_i, \ldots, y_n$ are realization of random variables $Y_1, \ldots, Y_n$. The joint distribution of $Y_1, \ldots, Y_n$ is denoted by ξ, which is called the superpopulation distribution. For example, we assume $\mathbb{E}_\xi[Y_i|\mathbf{x}_i] = \mathbf{x}_i^\top \beta$ and $\mathrm{V}_\xi(Y_i|\mathbf{x}_i) = \sigma_i^2$ for some unknown model parameter β and σ_i. The selected sample D is treated as a constant and the sample values of y_i are random. Estimation and inference are deduced conditional on the selected sample and the model.

For OPE, we have $y_{s,a} = P(s)\pi(a|s)r(s,a)$ and auxiliary vector $\mathbf{x}_{s,a} = P(s)\pi(a|s)\phi(s,a)$. We assume a linear model: $\mathbb{E}_\xi[Y_{s,a}|\mathbf{x}_{s,a}] = \mathbf{x}_{s,a}^\top \beta$, $\mathrm{V}_\xi(Y_{s,a}|\mathbf{x}_{s,a}) = \sigma_{s,a}^2 = (P(s)\pi(a|s)\sigma)^2$, and $Y_{s,a}$'s are independent. Using the WLS estimator to estimate β

$$\begin{aligned}\hat{\beta} &= \left(\sum_{(s,a)\in D} \frac{\mathbf{x}_{s,a}\mathbf{x}_{s,a}^\top}{\sigma_{s,a}^2}\right)^\dagger \left(\sum_{(s,a)\in D} \frac{\mathbf{x}_{s,a} y_{s,a}}{\sigma_{s,a}^2}\right) \\ &= \left(\sum_{(s,a)\in D} \phi(s,a)\phi(s,a)^\top\right)^\dagger \left(\sum_{(s,a)\in D} \phi(s,a) r(s,a)\right),\end{aligned}$$

we have the model-based estimator

$$\hat{t}_{\mathrm{MB}} = \sum_{(s,a)\in D} y_{s,a} + \sum_{(s,a)\notin D} \mathbf{x}_{s,a}^\top \hat{\beta}.$$

That is, the population total is estimated by the total of study variables in the sample and the total of the study variables of units not in the sample.

The model-based estimator is similar to the direct method (DM) in OPE. The key difference is that DM does not use the sample value of $y_{s,a}$ but uses

the prediction for all units, that is,

$$\hat{t}_{\mathrm{DM}} = \sum_{(s,a)\in\mathcal{U}} \mathbf{x}_{s,a}^{\top}\hat{\beta}.$$

The model-based survey sampling framework provide a way to do inference for the DM estimator, which is conditional on the selected sample and the model ξ. Let $t_{\mathbf{x}} = \sum_{(s,a)\in\mathcal{U}} \mathbf{x}_{s,a}$, then

$$\mathrm{V}_{\xi}(\hat{t}_{\mathrm{DM}}) = \mathrm{V}_{\xi}\left(\sum_{(s,a)\in\mathcal{U}} \mathbf{x}_{s,a}^{\top}\hat{\beta}\right) = t_{\mathbf{x}}^{\top}\mathrm{V}_{\xi}(\hat{\beta})t_{\mathbf{x}} = \sigma^2 t_{\mathbf{x}}^{\top}\left(\sum_{(s,a)\in D}\phi(s,a)\phi(s,a)^{\top}\right)^{\dagger} t_{\mathbf{x}}.$$

Plugging in the estimator $\hat{\sigma}^2 = \frac{1}{n-p}\sum_{(s,a)\in D}(r(s,a) - \phi(s,a)^{\top}\hat{\beta})^2$ for σ, we have an estimated variance

$$\hat{\mathrm{V}}(\hat{t}_{\mathrm{DM}}) = \hat{\sigma}^2 t_{\mathbf{x}}^{\top}\left(\sum_{(s,a)\in D}\phi(s,a)\phi(s,a)^{\top}\right)^{\dagger} t_{\mathbf{x}}.$$

C.2 Theoretical analysis

For the theoretical analysis, we make the following assumptions:

1. $\forall(s,a)\in\mathcal{U}$, $L_p \le p_{s,a}$ for some real number $L_p > 0$.
2. $\forall(s,a)\in\mathcal{U}$, $L_y \le y_{s.a} \le U_y$ for some real number L_y, U_y.
3. $\forall(s,a)\in\mathcal{U}$, $L_x \le \phi(s,a) \le U_x$ for some real vector L_x, U_x. The inequality holds element-wise.
4. The estimated matrix of the covariates $\sum_{(s,a)\in D}\frac{\pi(a|s)}{\pi_b(a|s)}\phi(s,a)\phi(s,a)^{\top}$ and the finite population matrix $\sum_{(s,a)\in\mathcal{U}}\phi(s,a)\phi(s,a)^{\top}$ are invertible.

In short, we need to make sure the data collection policy chooses each action with a non-zero probability, and the reward and feature vector are bounded.

C.2.1 Proof of Theorem 6

Definition 5 (The ratio estimator)**.** Let $z_{s,a} \in \mathbb{R}$ be the auxiliary variable, t_z be the populating total of the auxiliary variable, which is assumed to be

known, and $\hat{t}_{\text{HH}}$ and $\hat{t}_z$ be the HH estimator for t_y and t_z respectively. The ratio estimator is given by

$$\hat{t}_{\text{Ratio}} = t_z \frac{\hat{t}_{\text{HH}}}{\hat{t}_z}.$$

Now we show that the ratio estimator is a special case of the regression estimator.

Lemma 4. Suppose that we have univariate auxiliary information $z_{s,a}$. In the linear model $\mathbb{E}_\xi[Y_{s,a}] = \beta z_{s,a}$ and $\mathrm{V}_\xi(Y_{s,a}) = \sigma^2_{s,a} = z_{s,a}\sigma^2$ for some $\beta \in \mathbb{R}$ and $\sigma \in \mathbb{R}^+$, the regression estimator is equivalent to the ratio estimator.

Proof. First, note that the regression estimator has an alternative expression as

$$\begin{aligned}
\hat{t}_{\text{Reg}} &= \sum_{(s,a)\in\mathcal{U}} z_{s,a}\hat{\beta} + \sum_{(s,a)\in D} \frac{y_{s,a} - z_{s,a}\hat{\beta}}{np_{s,a}} \\
&= \sum_{(s,a)\in D} \frac{y_{s,a}}{np_{s,a}} + \left(\sum_{(s,a)\in\mathcal{U}} z_{s,a} - \sum_{(s,a)\in D} \frac{z_{s,a}}{np_{s,a}} \right) \hat{\beta} \\
&= \sum_{(s,a)\in D} \frac{y_{s,a}}{np_{s,a}} \\
&\quad + \left(\sum_{(s,a)\in\mathcal{U}} z_{s,a} - \sum_{(s,a)\in D} \frac{z_{s,a}}{np_{s,a}} \right) \left(\sum_{(s,a)\in D} \frac{z_{s,a}z_{s,a}}{np_{s,a}\sigma^2_{s,a}} \right)^{-1} \left(\sum_{(s,a)\in D} \frac{z_{s,a}y_{s,a}}{np_{s,a}\sigma^2_{s,a}} \right) \\
&= \sum_{(s,a)\in D} \frac{y_{s,a}}{np_{s,a}} \underbrace{\left[1 + \left(\sum_{(s,a)\in\mathcal{U}} z_{s,a} - \sum_{(s,a)\in D} \frac{z_{s,a}}{np_{s,a}} \right) \left(\sum_{(s,a)\in D} \frac{z_{s,a}z_{s,a}}{np_{s,a}\sigma^2_{s,a}} \right)^{-1} \frac{z_{s,a}}{\sigma^2_{s,a}} \right]}_{g_{s.a}} \\
&= \sum_{(s,a)\in D} \frac{g_{s,a}y_{s,a}}{np_{s,a}}
\end{aligned}$$

where $g_{s,a}$ can be viewed as the weight for each unit in the sample.

Under the model $\sigma^2_{s,a} = z_{s,a}\sigma^2$, for each $(s', a') \in D$, we have

$$\begin{aligned} g_{s',a'} &= 1 + \left(\sum_{(s,a)\in\mathcal{U}} z_{s,a} - \sum_{(s,a)\in D} \frac{z_{s,a}}{np_{s,a}}\right)\left(\sum_{(s,a)\in D} \frac{z_{s,a}z_{s,a}}{np_{s,a}\sigma^2 z_{s,a}}\right)^{-1}\left(\frac{z_{s',a'}}{\sigma^2 z_{s',a'}}\right) \\ &= 1 + \left(\sum_{(s,a)\in\mathcal{U}} z_{s,a} - \sum_{(s,a)\in D} \frac{z_{s,a}}{np_{s,a}}\right)\left(\sum_{(s,a)\in D} \frac{z_{s,a}}{np_{s,a}}\right)^{-1} \\ &= 1 + \left(\sum_{(s,a)\in\mathcal{U}} z_{s,a}\right)\left(\sum_{(s,a)\in D} \frac{z_{s,a}}{np_{s,a}}\right)^{-1} - 1 \\ &= t_z \Big/ \left(\sum_{(s,a)\in D} \frac{z_{s,a}}{np_{s,a}}\right) \end{aligned}$$

The second equality follows by cancelling out the right most term with the σ^2 in the inverse bracket. Note that the weight is the same for each unit in the sample. Plugging into the previous equation, we have

$$\hat{t}_{\text{Reg}} = t_z \frac{\sum_{(s,a)\in D} y_{s,a}/np_{s,a}}{\sum_{(s,a)\in D} z_{s,a}/np_{s,a}} = \hat{t}_{\text{Ratio}}.$$

□

Theorem 6 (WIS as a special case of the regression-assisted estimator)**.** Suppose we use a linear regression model with univariate feature $\phi(s, a) = 1$. Then the regression-assisted DR estimator with estimated coefficient $\hat{\beta}$ from Eq (6.3) has the same form as the WIS estimator:

$$\hat{t}_{\text{Reg}} = \sum_{(s,a)\in D} \frac{\pi(a|s)/\pi_b(a|s)}{\sum_{(s',a')\in S} \pi(a'|s')/\pi_b(a'|s')} r(s, a). \tag{6.5}$$

Proof. We first show that the WIS estimator belongs to a class of estimators called the ratio estimator in survey sampling in Definition 5. Suppose the auxiliary variable $z_{s,a} = P(s)\pi(a|s)$, and we know $t_z = \sum_{(s,a)\in\mathcal{U}} P(s)\pi(a|s) = 1$. Then, the ratio estimator is

$$\begin{aligned} \hat{t}_{\text{Ratio}} = t_z \frac{\hat{t}_{\text{HH}}}{\hat{t}_z} &= \left(\sum_{(s,a)\in D} \frac{y_{s,a}}{np_{s,a}}\right)\left(\sum_{(s,a)\in D} \frac{z_{s,a}}{np_{s,a}}\right)^{-1} \\ &= \sum_{(s,a)\in D} \frac{\pi(a|s)/\pi_b(a|s)}{\sum_{(s',a')\in D} \pi(a'|s')/\pi_b(a'|s')} r(s, a) \end{aligned}$$

which is the WIS estimator in the OPE literature.

Then, we prove a more general statement that a ratio estimator with univariate auxiliary information $z_{s,a}$ is a special case of the regression estimator under the linear model $\mathbb{E}_\xi[Y_{s,a}] = \beta z_{s,a}$ and $\mathrm{V}_\xi(Y_{s,a}) = \sigma_{s,a}^2 = z_{s,a}\sigma^2$ for some $\beta \in \mathbb{R}$ and $\sigma \in \mathbb{R}^+$ in Lemma 4. □

C.2.2 Proof of Theorem 7

Lemma 5 (Variance of the HH estimator for multinomial design)**.** Let z be a mapping from $\mathcal{S} \times \mathcal{A}$ to $[a, b]$ for two constants $a < b$, and $\hat{t}_{\mathrm{HH}}$ be the HH estimator for the variable $y_{s,a} = P(s)\pi(a|s)z(s,a)$. With multinomial design n and $p_{s,a} = P(s)\pi_b(a|s)$, the variance is given by

$$\mathrm{V}(\hat{t}_{\mathrm{HH}}) = \frac{1}{n}\left(\sum_{(s,a)\in U}\frac{y_{s,a}^2}{p_{s,a}} - t_y^2\right) = \frac{1}{n}\left(\sum_{(s,a)\in U} P(s)\pi(a|s)\frac{\pi(a|s)}{\pi_b(a|s)}z(s,a)^2 - t_y^2\right).$$

Proof. Recall the HH estimator is

$$\hat{t}_{\mathrm{HH}} = \sum_D \frac{y_{s,a}}{np_{s,a}} = \sum_{\mathcal{U}} \frac{I_{s,a}y_{s,a}}{np_{s,a}}.$$

where $I_{s,a}$ is the (s,a)-th element of the design vector $\mathbf{I}$. The variance is

$$\begin{aligned}\mathrm{V}(\hat{t}_{\mathrm{HH}}) &= \mathrm{V}\left(\sum_{\mathcal{U}} I_{s,a}\frac{y_{s,a}}{np_{s,a}}\right) \\ &= \sum_{(s,a)\in\mathcal{U}} \mathrm{V}(I_{s,a})\left(\frac{y_{s,a}}{np_{s,a}}\right)^2 + \sum_{(s,a)\neq(s',a')} \mathrm{Cov}(I_{s,a}, I_{s',a'})\left(\frac{y_{s,a}}{np_{s,a}}\right)\left(\frac{y_{s',a'}}{np_{s',a'}}\right).\end{aligned}$$

We know $\mathrm{V}(I_{s,a}) = np_{s,a}(1-p_{s.a})$ and $\mathrm{Cov}(I_{s,a}, I_{s',a'}) = -np_{s.a}p_{s',a'}$ from the properties of the multinomial distribution, hence, after some calculation, we have $\mathrm{V}(\hat{t}_{\mathrm{HH}}) = \frac{1}{n}\left(\sum_{(s,a)\in\mathcal{U}}\frac{y_{s,a}^2}{p_{s,a}} - t_y^2\right)$. The proof is completed by plugging in the value of $y_{s,a}$ and $p_{s,a}$. □

Theorem 7 (Properties of the estimator)**.** Let $\mathrm{AV}(\cdot)$ denote the asymptotic variance in term of the first order, that is $\mathrm{V}(\cdot) = \mathrm{AV}(\cdot) + o(n^{-1})$, we have (1) $\hat{t}_{\mathrm{Reg}}$ is asymptotically unbiased with a bias of order $O(n^{-1})$, and (2)

$$\mathrm{AV}(\hat{t}_{\mathrm{Reg}}) = \frac{1}{n}\left(\sum_{(s,a)\in U} P(s)\frac{\pi(a|s)^2}{\pi_b(a|s)}(r(s,a) - \phi(s,a)^\top\beta)^2 - t_e^2\right)$$

where $t_e = \sum_{\mathcal{U}} P(s)\pi(a|s)(r(s,a) - \phi(s,a)^\top\beta)$.

Proof. Note that we assume the first term in Eq (6.3) is invertible. We can write the estimator as $\hat{t}_{\text{Reg}} = \hat{t}_y + (t_z - \hat{t}_z)\hat{A}^{-1}\hat{C}$ where $\hat{t}_y$ is the HH estimator for $t_y = \sum_{\mathcal{U}} P(s)\pi(a|s)r(s,a)$ and $t_z = \sum_{\mathcal{U}} \mathbf{z}_{s,a}$ and $\hat{t}_z$ be the HH estimator for t_z. $\hat{A}$ and $\hat{C}$ denote the first and second matrix is Eq (6.3) respectively. Moreover, let t_{zj} be the j-th element of the vector t_z, and $\hat{t}_{zj}$ be the j-th element of the vector $\hat{t}_{zj}$.

Let $A = \sum_{\mathcal{U}} P(s)\pi(a|s)\phi(s,a)\phi(s,a)^\top$, $C = \sum_{\mathcal{U}} P(s)\pi(a|s)\phi(s,a)r(s,a)$ and $B = A^{-1}C$. Using the Taylor linearization technique (see Section 6.6 of Särndal et al., 1992), and we can approximate $\hat{t}_{\text{Reg}}$ at $\hat{t}_y = t_y$, $\hat{t}_1 = t_1$, $\hat{t}_z = t_z$, $\hat{A} = A$ and $\hat{C} = C$:

$$\begin{aligned}
\hat{t}_{\text{Reg}} &= t_y + 1(\hat{t}_y - t_y) - \sum_j B_j(\hat{t}_{z,j} - t_{z,j}) \\
&\quad + \sum_{i,j}(t_z - t_z)^\top[-A^{-1}E_{ij}A^{-1}]C(\hat{A}_{ij} - A_{ij}) \\
&\quad + \sum_j (t_z - \hat{t}_z)^\top e_j(\hat{C}_j - C_j) + \dots \\
&= \hat{t}_y + (t_z - \hat{t}_z)^\top B + ...
\end{aligned}$$

where E_{ij} is a matrix where the ij- and ji-th elements are one and all other elements are zero, and e_j is a vector where the j-th element is one and zero otherwise.

Since the random variable is bounded, the moments exist. Taking the expectation, we get

$$\mathbb{E}[\hat{t}_{\text{Reg}}] = \mathbb{E}[\hat{t}_y + (t_z - \hat{t}_z)^\top B] + O(n^{-1}) = t_y + O(n^{-1}). \tag{C.1}$$

The first equality follows from the remainder terms of the Taylor expansion are the expectations of $(\hat{t}_y - t_y)^p$ and $(\hat{t}_{z,j} - t_{z,j})^p$ for $p \geq 2$, which is of order $O(1/n)$. The second equality follows from $\hat{t}_z$ is an unbiased estimator for t_z. Therefore, $\hat{t}_{\text{Reg}}$ is asymptotically unbiased.

Furthermore,

$$\begin{aligned}
\mathrm{V}(\hat{t}_{\mathrm{Reg}}) &= \mathbb{E}[(\hat{t}_{\mathrm{Reg}} - \mathbb{E}[\hat{t}_{\mathrm{Reg}}])^2] \\
&= \mathbb{E}[(\hat{t}_{\mathrm{Reg}} - t_y + t_y - \mathbb{E}[\hat{t}_{\mathrm{Reg}}])^2] \\
&= \mathbb{E}[(\hat{t}_{\mathrm{Reg}} - t_y)^2 + (t_y - \mathbb{E}[\hat{t}_{\mathrm{Reg}}])^2 + 2(\hat{t}_{\mathrm{Reg}} - t_y)(t_y - \mathbb{E}[\hat{t}_{\mathrm{Reg}}])] \\
&= \mathbb{E}[(\hat{t}_{\mathrm{Reg}} - t_y)^2] + o(n^{-1}) \\
&= \mathbb{E}[(\underbrace{\hat{t}_y - \hat{t}_z B}_{(a)} - t_y + t_x B)^2] + o(n^{-1})
\end{aligned}$$

The last equality comes from Eq (C.1). Note that (a) is the HH estimator $\hat{t}_e = \sum_D \frac{\pi(a|s)}{n\pi_b(a|s)}(r(s,a) - \phi(s,a)^\top B)$ so the expectation (the first term in the last line) is the variance of (a) which is given by

$$\frac{1}{n}\left(\sum_{(s,a)\in U} P(s)\pi(a|s)\frac{\pi(a|s)}{\pi_b(a|s)}(r(s,a) - \phi(s,a)^\top B)^2 - t_e^2\right)$$

with $t_e = \sum_{\mathcal{U}} P(s)\pi(a|s)(r(s,a) - \phi(s,a)^\top B)$ by Lemma 5.

Since the variance converges to zero and the estimator is asymptotically unbiased, we also know $\hat{t}_{\mathrm{Reg}} \xrightarrow{p} t_y$. □

C.2.3 Proof of Theorem 8

Proof of consistency. Define

$$\hat{\mathrm{V}}_n(\beta) = \frac{1}{n(n-1)}\left[\sum_{(s,a)\in D}\left(\frac{\pi(a|s)}{\pi_b(a|s)}(r(s,a) - \phi(s,a)^\top\beta)\right)^2 - n\hat{t}_e(\beta)^2\right], \text{ and}$$

$$\mathrm{V}_n(\beta) = \frac{1}{n}\left(\sum_{(s,a)\in U} P(s)\pi(a|s)\frac{\pi(a|s)}{\pi_b(a|s)}(r(s,a) - \phi(s,a)^\top\beta)^2 - t_e(\beta)^2\right)$$

where $\hat{t}_e(\beta) = \sum_D \frac{\pi(a|s)}{n\pi_b(a|s)}(r(s,a) - \phi(s,a)^\top\beta)$ and $t_e(\beta) = \sum_{\mathcal{U}} P(s)\pi(a|s)(r(s,a) - \phi(s,a)^\top\beta)$. Then it is sufficient to show that

$$n|\hat{\mathrm{V}}_n(\hat{\beta}_n) - \mathrm{V}_n(\beta_{WLS})| \xrightarrow{p} 0.$$

For $\epsilon > 0$, by the triangle inequality, we have

$$\begin{aligned}
&\Pr(n|\hat{\mathrm{V}}_n(\hat{\beta}_n) - \mathrm{V}_n(\beta_{WLS})| > \epsilon) \\
&\leq \Pr(n|\hat{\mathrm{V}}_n(\hat{\beta}_n) - \hat{\mathrm{V}}_n(\beta_{WLS})| > \epsilon/2) + \Pr(n|\hat{\mathrm{V}}_n(\beta_{WLS}) - \mathrm{V}_n(\beta_{WLS})| > \epsilon/2).
\end{aligned}$$

For the first term, using the fact that $\hat{\beta}_n \xrightarrow{p} \beta_{WLS}$ and the continuous mapping theorem, we get $\hat{\mathrm{V}}_n(\hat{\beta}_n) \xrightarrow{p} \hat{\mathrm{V}}_n(\beta_{WLS})$, which implies $\lim_{n\to\infty} \Pr(n|\hat{\mathrm{V}}_n(\hat{\beta}_n) - \hat{\mathrm{V}}_n(\beta_{WLS})| > \epsilon/2) = 0$.

Define $e_{s,a}(\beta) = r(s,a) - \phi(s,a)^\top \beta$, $t_{we^2}(\beta) = \sum_{\mathcal{U}} P(s)\pi(a|s)\frac{\pi(a|s)}{\pi_b(a|s)} e_{s,a}(\beta)^2$ (the first term of $\mathrm{V}_n(\beta)$) and $\hat{t}_{we^2}(\beta) = \sum_D \left(\frac{\pi(a|s)}{n\pi_b(a|s)} e_{s,a}(\beta)\right)^2$ (the first term of $\hat{\mathrm{V}}(\beta)$). Then, for the second term, we have

$$
\begin{aligned}
&\Pr(n\left|\hat{\mathrm{V}}_n(\beta_{WLS}) - \mathrm{V}_n(\beta)\right| > \epsilon/2) \\
&= \Pr(\left|\frac{n}{n-1}\hat{t}_{we^2}(\beta_{WLS}) - \frac{n}{n-1}\hat{t}_e(\beta_{WLS})^2 - t_{we^2}(\beta_{WLS}) + t_e(\beta_{WLS})^2\right| > \epsilon/2) \\
&\leq \Pr(\left|\frac{n}{n-1}\hat{t}_{we^2}(\beta_{WLS}) - t_{we^2}(\beta_{WLS})\right| > \epsilon/4) \\
&\quad + \Pr(\left|\frac{n}{n-1}\hat{t}_e(\beta_{WLS})^2 - t_e(\beta_{WLS})^2\right| < \epsilon/4).
\end{aligned}
$$

Note that $\hat{t}_{we^2}(\beta_{WLS})$ and $\hat{t}(\beta_{WLS})^2$ are the HH estimators for $t_{we^2}(\beta_{WLS})$ and $t_e(\beta_{WLS})$ respectively, so they are consistent. As a result, we have

$$
\lim_{n\to\infty} \Pr(n\left|\hat{\mathrm{V}}_n(\hat{\beta}_{WLS}) - \mathrm{V}_n(\beta_{WLS})\right| > \epsilon/2) = 0,
$$

which completes the proof. □

Proof of asymptotic normality. It is known that the HH estimator for with-replacement sampling is asymptotically normal (for example, see Theorem 2 of Félix-Medina, 2003 or K. McConville, 2011), that is,

$$
\begin{bmatrix} \sqrt{n}(\hat{t}_y - t_y) \\ \sqrt{n}(\hat{t}_z - t_z) \end{bmatrix} \xrightarrow{D} \mathcal{N}\left(0, \begin{bmatrix} \Sigma^y & \mathbf{\Sigma}^{yz} \\ \mathbf{\Sigma}^{zy} & \mathbf{\Sigma}^z \end{bmatrix}\right)
$$

where $\Sigma^y, \mathbf{\Sigma}^{yz}, \mathbf{\Sigma}^{zy}$ and $\mathbf{\Sigma}^z$ are the limiting covariance matrices. Then we follow the proof idea from Theorem 3.2 of K. McConville, 2011. Using the Slutsky's Theorem and the fact that $\hat{\beta}_n \xrightarrow{p} \beta_{WLS}$, we have

$$
\begin{bmatrix} \sqrt{n}(\hat{t}_y - t_y) \\ \sqrt{n}(\hat{t}_z - t_z)\hat{\beta}_n \end{bmatrix} \xrightarrow{D} \mathcal{N}(0, \begin{bmatrix} \Sigma^y & \mathbf{\Sigma}^{yz}\beta_{WLS} \\ \beta_{WLS}^\top \mathbf{\Sigma}^{zy} & \beta_{WLS}^\top \mathbf{\Sigma}^z \beta_{WLS} \end{bmatrix}).
$$

Note that $\sqrt{n}(\hat{t}_{\mathrm{Reg}} - t_y) = \sqrt{n}(\hat{t}_y - t_y) - \sqrt{n}(\hat{t}_z - t_z)\hat{\beta}_n$. By the Delta method, we have $\sqrt{n}(\hat{t}_{\mathrm{Reg}} - t_y) \xrightarrow{D} \mathcal{N}(0, \sigma^2)$ where $\sigma^2 = \Sigma^y - \Sigma^{yz}\beta_{WLS} - \beta_{WLS}^\top \Sigma^{zy} + \beta_{WLS}^\top \Sigma^z \beta_{WLS}$. Note that we can write the variance of $\hat{t}_y - \hat{t}_z\beta_{WLS}$ as $\mathrm{V}(\hat{t}_y -$

$\hat{t}_z\beta_{WLS}) = \frac{1}{n}(\Sigma^y - \Sigma^{yz}\beta_{WLS} - \beta_{WLS}^\top\Sigma^{zy} + \beta_{WLS}^\top\Sigma^z\beta_{WLS})$, and in the proof for Theorem 7, we show that the asymptotic variance of $\hat{t}_{\text{Reg}}$ is $\text{AV}(\hat{t}_{\text{Reg}}) = \text{V}(\hat{t}_y - \hat{t}_z\beta_{WLS})$. Therefore, $\text{AV}(\hat{t}_{\text{Reg}}) = \sigma^2/n$, and $(\hat{t}_{\text{Reg}} - t_y)/\sqrt{\text{AV}(\hat{t}_{\text{Reg}})} \xrightarrow{D} \mathcal{N}(0,1)$.

By the consistency of the variance estimator and Slutsky's theorem, we have

$$\frac{\hat{t}_{\text{Reg}} - t_y}{\sqrt{\hat{\text{V}}(\hat{t}_{\text{Reg}})}} = \frac{\hat{t}_{\text{Reg}} - t_y}{\sqrt{\text{AV}(\hat{t}_{\text{Reg}})}} \frac{\sqrt{\text{AV}(\hat{t}_{\text{Reg}})}}{\sqrt{\hat{\text{V}}(\hat{t}_{\text{Reg}})}} \xrightarrow{D} \mathcal{N}(0,1).$$

□

C.3 Experiment details

For the semi-synthetic datase, we generate a sequence of reward functions based on the non-stationary recommendation environment used in Chandak et al., 2020. For each positive context-action pair in the original classification dataset, the reward follows a sine wave with noises: $r_k(s,a) = 0.5 + \text{amplitude}_{s,a} * \sin(k * \text{frequency}_{s,a}) + 0.01\varepsilon$ where $\varepsilon \sim \text{Unif}([0,1])$. For each interval, we also randomly sample some context-action pairs and set their rewards to positive random values to increase the noise.

To obtain a target policy for the Youtube and MovieLens dataset, we fist train a classifier on a small subset of the original multi-label classification dataset. Then we apply the softmax function on the outputs of the trained classifiers to obtain a probability distribution over actions for each context. The conditional distribution is used as the target policy.

Similarly for the RL environment, the reward follows $r_k(s,a) = \mu_{s,a} + 0.25 * \sin(k * \text{frequency}_{s,a}) + 0.01 * \varepsilon$ where $\varepsilon \sim \text{Unif}([0,1])$. To obtain a target policy, we fist train a Q-learning agent on the underlying environment for 1000 episodes and then apply the softmax function on the Q-value as the target policy.