Value Enhancement of Reinforcement Learning via Efficient and Robust Trust Region Optimization

Chengchun Shi, Zhengling Qi, Jianing Wang & Fan Zhou

To cite this article: Chengchun Shi, Zhengling Qi, Jianing Wang & Fan Zhou (2024) Value Enhancement of Reinforcement Learning via Efficient and Robust Trust Region Optimization, Journal of the American Statistical Association, 119:547, 2011-2025, DOI: 10.1080/01621459.2023.2238942

To link to this article: https://doi.org/10.1080/01621459.2023.2238942

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2024, VOL. 119, NO. 547, 2011–2025: Theory and Methods
https://doi.org/10.1080/01621459.2023.2238942

Value Enhancement of Reinforcement Learning via Efficient and Robust Trust Region Optimization

Chengchun Shi[a], Zhengling Qi[b], Jianing Wang[c], and Fan Zhou[c]

[a]Department of Statistics, London School of Economics and Political Science, London, UK; [b]Department of Decision Sciences, The George Washington University, Washington, DC; [c]School of Statistics and Management, Shanghai University of Finance and Economics, Shanghai, China

ABSTRACT

Reinforcement learning (RL) is a powerful machine learning technique that enables an intelligent agent to learn an optimal policy that maximizes the cumulative rewards in sequential decision making. Most of methods in the existing literature are developed in *online* settings where the data are easy to collect or simulate. Motivated by high stake domains such as mobile health studies with limited and pre-collected data, in this article, we study *offline* reinforcement learning methods. To efficiently use these datasets for policy optimization, we propose a novel value enhancement method to improve the performance of a given initial policy computed by existing state-of-the-art RL algorithms. Specifically, when the initial policy is not consistent, our method will output a policy whose value is no worse and often better than that of the initial policy. When the initial policy is consistent, under some mild conditions, our method will yield a policy whose value converges to the optimal one at a faster rate than the initial policy, achieving the desired "value enhancement" property. The proposed method is generally applicable to any parameterized policy that belongs to certain pre-specified function class (e.g., deep neural networks). Extensive numerical studies are conducted to demonstrate the superior performance of our method. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received August 2021
Accepted July 2023

KEYWORDS
Mobile health studies; Offline reinforcement learning; Semi-parametric efficiency; Trust region optimization

1. Introduction

Reinforcement learning (RL, see e.g., Sutton and Barto 2018) is concerned with how agents take sequential actions in dynamic environments, with the main goal of maximizing the cumulative rewards they receive. In recent years, we have seen tremendous achievements of RL in artificial intelligence (AI). For example, *AlphaGo* (Silver et al. 2016), one of the most successful applications in AI, makes use of reinforcement learning and deep learning algorithms for teaching machines to play the board game called Go, and has beaten many top human players. The appealing performance of RL has also been demonstrated in many scientific fields. In medical applications, RL has been used to help clinicians make better treatment decisions for patients with sepsis (Komorowski et al. 2018). In economics, econometricians often study dynamic discrete choice models (Rust 1987) in order to understand the behavior of rational agents, which is similar to the inverse RL problem (Abbeel and Ng 2004). In operations research, RL has been widely applied to business operations such as supply chain management, finance and logistics (Hubbs et al. 2020). For an overview of various applications of RL, we refer to Section 5 of Li (2017).

Our research in this article is partly motivated by recently emerging mobile health (mHealth) studies. Advancements in mobile and sensor technologies provides us with a unique opportunity to deliver health interventions at anytime and anywhere for promoting healthy behaviors such as regular physical activities and preventing drug abuse, etc. For example, the OhioT1DM dataset (Marcolino et al. 2018) was developed for promoting blood glucose level prediction in order to improve the health and wellbeing of people with type 1 diabetes. It contains data information of 6 people for 8 weeks. For each patient, their treatment information was collected during insulin pump therapy with continuous glucose monitoring (CGM). In addition, blood glucose levels and self-reported times of meals and exercises were also constantly measured and recorded via a custom smartphone app. Finding an optimal insulin pumping policy for each patient at different scenarios may potentially improve their health status (Shi et al. 2020b). This matches the goal of RL algorithms.

A fundamental question we aim to investigate here is how to learn an optimal policy efficiently from the batch data in high-stake domains such as mHealth. Solving this question faces at least two major challenges. First, different from the standard clinical trial data, mobile health data usually consist of a large number of decision points for each patient but the number of patients may be limited (e.g., in OhioT1DM dataset, 6 patients with a few thousands decision points). This posits a unique challenge for searching an optimal policy. In statistics, there is a rich literature in studying dynamic treatment regimes (DTR, see e.g., Murphy 2003; Qian and Murphy 2011; Chakraborty and Moodie 2013; Zhao et al. 2015; Shi et al. 2018; Wang et al. 2018).

CONTACT Zhengling Qi *qizhengling@gwu.edu* Department of Decision Sciences, The George Washington University, Washington, DC; Fan Zhou *zhoufan@mail.shufe.edu.cn* School of Statistics and Management, Shanghai University of Finance and Economics, Shanghai, China.
The first two authors contribute equally to this article.

Supplementary materials for this article are available online. Please go to *www.tandfonline.com/r/JASA*.

For a review of DTR, see Laber et al. (2014), Kosorok and Laber (2019), and Tsiatis et al. (2019). However, these methods are mainly designed for only a few treatment decision points and often require a large number of patients in the observed data in order to be consistent.

Second, different from online RL domains such as video games, where actively interacting with the environment is feasible and data are easy to generate or simulate, in some high stake domains, data are often pre-collected according to some experimental design and very limited. With such limited data, it is essential to study how to efficiently learn the optimal policy from the batch data. We remark that the main focus of RL in the computer science literature is for online learning. Among all the methods available, Q-learning is arguably the most popular model-free RL algorithms (Watkins and Dayan 1992). It derives the optimal policy by learning an optimal Q-function (see the definition of Q-function in Section 2.2). Follow this line of research, variants of Q-learning methods have been proposed including the fitted Q-iteration (FQI, Ernst et al. 2005; Fan et al. 2020a), deep Q-network (DQN, Mnih et al. 2015), among many others. Policy-based learning is another class of RL algorithms that searches the optimal policy among a parameterized policy class. Some popular algorithms include REINFORCE and actor-critic methods (see e.g., Sutton and Barto 2018, chap. 13). Since these algorithms are primarily motivated by the application of developing artificial intelligence in online video games, their generalization to offline settings such as mobile health applications remain largely unexplored.

To address the first challenge, we model the observed data by a time-homogeneous Markov decision process (MDP Puterman 1994). This framework is particularly suitable to model the data collected from mobile health studies where the total number of decision points are often large (see e.g., Liao, Klasnja, and Murphy 2019; Liao, Qi, and Murphy 2020). The assumptions of Markov and time homogeneity enable a consistent estimation of the optimal policy even with only a few patients.

To address the second challenge, we develop a novel procedure to derive the optimal policy. Recently, a few algorithms have been developed in the statistics literature for policy optimization in mHealth applications (Ertefaie and Strawderman 2018; Luckett et al. 2020; Liao, Qi, and Murphy 2020; Hu et al. 2021). In particular, Liao, Qi, and Murphy (2020) proposed a statistically efficient batch policy learning method under the average reward MDP. However, due to the policy dependent structure of nuisance functions such as Q-function and the marginal density ratio, their proposed algorithm is computational inefficient as it requires updating the nuisance functions estimation in each iteration of their policy gradient decent algorithm. Instead of proposing a specific algorithm for policy optimization, we devise a "value enhancement" method that is generally applicable to any given initial policy computed by some state-of-the-art RL algorithm to improve their performance. Basically, after employing some computational efficient RL algorithm and obtaining an initial policy, we take a one-step update of this policy via efficiently estimating the value enhancement component (defined in Section 2.3) and solving a constrained optimization problem, thus, taking advantage of computational efficiency from existing state-of-art RL algorithms without requiring iteratively updating the nuisance functions. More importantly, the proposed procedure guarantees that when the initial policy is not consistent, the output policy by the proposed algorithm is no worse and often better than the initial policy. If consistent, our method will yield a policy whose value converges to the optimal one at a faster rate, achieving the desired "value enhancement" property. Recently, in the computer science literature, Kallus and Uehara (2020) developed an offline policy gradient algorithm that considered statistically efficient estimation of the policy gradient. Our proposal differs from theirs in that we focus on developing a general value enhancement tool that is applicable to any existing RL algorithms to improve their performances.

Our method is inspired by Lemma 6.1 in Kakade and Langford (2002) and the trust region policy optimization algorithm by Schulman et al. (2015), which was originally designed for the online setting. A key observation is that, the value difference between any two policies can be decomposed into a first-order component and a higher-order remainder term. The higher-order term can be lower bounded, based on which a minorization function can be constructed for the value function of any policy. One big advantage of working with this minorization function is that it intrinsically disentangles the policy-dependent structure of nuisance functions. This ensures the computational efficiency of the proposed algorithm.

The key "value enhancement" property relies crucially on statistically efficient estimation of the first-order term in the decomposition. In online settings, Schulman et al. (2015) proposed to simulate data trajectories to estimate this quantity. However, in offline settings, it remains unknown how to effectively evaluate this quantity based on the observed data. By leveraging semi-parametric statistics, we develop a triply robust estimator for the first-order term that is shown to achieve the efficiency bound when compared to the initial policy. By optimizing the proposed estimator, we are able to improve the value of the initial policy. The triply robustness property guarantees that the "value enhancement" property holds even when some nuisance function models are misspecified. The semi-parametric efficiency guarantees that the value can be enhanced at a sufficiently fast rate. This ensures the statistical efficiency of the proposed algorithm, which is necessary in the offline setting.

In theory, we establish the value enhancement property under mild conditions on the nuisance function estimators. In particular, we only require them to converge at a nonparametric rate. See Section 4 for details. This nice property is achieved mainly due to the innovative way we put together these nuisance function estimators, which leads to the triply-robust estimator with a parametric convergence rate for the first-order term. In addition, we remark that all our theoretical results related to estimation are established in terms of total decision points, thus, showing the proposed method is generally applicable even when the number of trajectories is small but the length of each trajectory is large, which is commonly seen in the mobile health applications.

The rest of this article is organized as follows. In Section 2, we introduce the offline RL problem in the framework of a time-homogeneous MDP and review the trust region algorithm. In Section 3, we present our value enhanced policy optimization method and the related estimation. In Section 4, we study statistical properties of our algorithm. In Section 5, extensive numerical studies including a toy example demonstrating the value

enhancement property, a real-data driven simulation study and a real data application are conducted to demonstrate the superior performance of the proposed method. Finally, we conclude our article in Section 6. All technical proofs and details can be found in the Supplementary Material.

2. Preliminaries

This section is organized as follows. We first introduce the offline data structure and describe the model setup in Section 2.1. In Section 2.2, we introduce some notations needed to derive our method. In Section 2.3, we review the trust region policy optimization (TRPO) method proposed by Schulman et al. (2015) for online learning.

2.1. Value Function and the Optimal Policy

Consider a single trajectory $\{(S_t, A_t, R_t)\}_{t\geq 0}$ where (S_t, A_t, R_t) denotes the state-action-reward triplet collected at time t. We use $\mathcal{S}$ and $\mathcal{A}$ to denote the state and action space, respectively. We assume $\mathcal{S}$ and $\mathcal{A}$ are discrete, and rewards R_t are uniformly bounded. The discrete state space assumption is imposed only to simplify the presentation and the theoretical analysis. Our proposed method is equally applicable to settings with continuous state space as well. The observed data consist of N trajectories, corresponding to N independent and identically distributed copies of $\{(S_t, A_t, R_t)\}_{t\geq 0}$. For any $i = 1, \ldots, N$, data collected from the ith trajectory can be summarized by $\{(S_{i,t}, A_{i,t}, R_{i,t}, S_{i,t+1})\}_{0\leq t<T_i}$, where T_i denotes the termination time for the ith trajectory.

A policy defines the agent's way of choosing the action at each decision time. A history-dependent policy π is a sequence of decision rules $\{\pi_t\}_{t\geq 0}$ such that each π_t maps the observed data history $\bar{S}_t = S_t \cup \{S_j, A_j, R_j\}_{0\leq j<t}$ to a probability mass function on $\mathcal{A}$, denoted by $\pi_t(\cdot|\bar{S}_t)$. When each π_t outputs a value in $\mathcal{A}$, π is referred to as a deterministic policy. Under π, the agent will set $A_t = a$ with probability $\pi_t(a|\bar{S}_t)$ at the tth decision point. Suppose there exists some function $\tilde{\pi}$ such that $\pi_t(\cdot|\bar{S}_t) = \tilde{\pi}(\cdot|S_t)$ almost surely for any t, then π is referred to as a stationary policy.

The primary goal of RL is to identify an optimal policy that maximizes the cumulative reward that the agent receives. To formally state this objective, we define the value function

$$V^{\pi}(s) = \sum_{t=0}^{+\infty} \gamma^t \mathbb{E}^{\pi}(R_t|S_0 = s), \tag{1}$$

where $\mathbb{E}^{\pi}$ denotes the expectation assuming the actions are selected according to π, and $0 \leq \gamma < 1$ denotes some discounted factor that balances the tradeoff between immediate and future rewards. We aim to learn a policy that maximizes the following integrated value function,

$$\mathcal{V}(\pi) = \sum_{s\in\mathcal{S}} V^{\pi}(s)\nu(s), \tag{2}$$

where ν denotes some *known* reference distribution function on $\mathcal{S}$. The known of reference distribution is a typical assumption in RL literature. We assume $\nu(s)$ is uniformly bounded away from zero for any $s \in \mathcal{S}$. When N is large, one may alternatively set ν to the distribution function of S_0 and estimate it via the empirical distribution function of the data samples $\{S_{i,0}\}_{1\leq i\leq N}$.

The following assumptions allow us to focus on stationary policies and serve as the foundations of the existing state-of-the-art RL algorithms:

(A1) *Markov assumption with stationary transitions*: there exists a Markov transition function p such that for any $t \geq 0$, $a \in \mathcal{A}$ and $s, s' \in \mathcal{S}$,

$$\Pr(S_{t+1} = s'|A_t = a, S_t = s, \{S_j, A_j, R_j\}_{0\leq j<t}) = p(s'|a, s).$$

(A2) *Conditional mean independence assumption with stationary reward functions*: there exists some function r such that for any $t \geq 0$, $s \in \mathcal{S}$ and $a \in \mathcal{A}$,

$$\begin{aligned}&\mathbb{E}(R_t|S_t = s, A_t = a, \{S_j, A_j, R_j\}_{0\leq j<t})\\ &\quad = \mathbb{E}(R_t|S_t = s, A_t = a) = r(s, a).\end{aligned}$$

These two assumptions require the future state and the conditional mean of the immediate reward to be independent of the past observations given the current state-action pair at each decision time t. Under these assumptions, there exists an optimal stationary policy π^{opt} whose value function $V^{\pi^{\text{opt}}}(s)$ is no worse than $V^{\pi}(s)$ for any history-dependent policy π and any $s \in \mathcal{S}$ (Puterman 1994, sec. 6.2). Consequently, it also maximizes the integrated value function $\mathcal{V}(\pi)$. (A1) and (A2) are testable from the observed data. See the goodness-of-fit test proposed by Shi et al. (2020a). In practice, to ensure the Markov property satisfied, we can construct the state by concatenating measurements over multiple decision points till the Markov property is satisfied (see Section 5.2 for details). To guarantee the transition probability p and the reward function r are time-homogeneous, we can include some auxiliary variables (e.g., time of the day) in the state. Hereafter, we restrict our attentions to the class of stationary policies that belong to certain pre-specified class Π (e.g., linear or neural networks). For any $\pi \in \Pi$, we use $\pi(\bullet|s)$ to denote the probability mass function on $\mathcal{A}$ that the agent will follow when the state value is s. We aim to learn $\pi^* \in \text{argmax}_{\pi\in\Pi}\mathcal{V}(\pi)$ based on the observed data.

To conclude this section, we impose one additional assumption that is commonly assumed in the literature to handle offline data (Sutton and Barto 2018):

(A3) The data are generated by some fixed stationary policy denoted by b. In addition, the stochastic process $\{(A_t, S_t)\}_{t\geq 0}$ is stationary.

Under (A1) and (A3), the process $\{(A_t, S_t)\}_{t\geq 0}$ forms a time-homogeneous Markov chain. We use p_∞ to denote the stationary distribution of the state-action pair. Suppose $p_\infty(a, s)$ is uniformly bounded away from zero for any $a \in \mathcal{A}, s \in \mathcal{S}$. The stationarity of $\{(A_t, S_t)\}_{t\geq 0}$ is assumed for convenience, since the Markov chain will eventually reach stationarity. For the ease of presentation, throughout this article, we assume (A1)-(A3) hold.

2.2. Additional Notations

For any $a \in \mathcal{A}$, $s \in \mathcal{S}$, we define the following action-value function associated with a given policy π as $Q^{\pi}(a, s) = \sum_{t=0}^{+\infty} \gamma^t \mathbb{E}^{\pi}(R_t|A_0 = a, S_0 = s)$, better known as the Q-function. By definition, it is equal to the discounted cumulative reward the agent receives when the initial action-state pair

equals (a, s) and all subsequent actions follow π. By (1), we have $V^\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s)Q^\pi(a, s)$ for any π.

The advantage function A^π over $\mathcal{A}\times\mathcal{S}$ associated with π is defined by the difference between the Q-function and the value function, that is, $A^\pi(a, s) = Q^\pi(a, s) - V^\pi(s)$ for any $a \in \mathcal{A}$ and $s \in \mathcal{S}$. It represents the gain of the expected cumulative reward by performing the action a rather than following π at the initial stage. By definition, we obtain

$$\sum_{a\in\mathcal{A}} \pi(a|s)A^\pi(a, s) = 0, \quad \text{for any } s, \pi. \tag{3}$$

We next introduce the discounted visitation probability. For any $t \geq 0$, let $p_t^\pi(s'|a, s)$ denote the t-step visitation probability $\Pr^\pi(S_t = s'|A_0 = a, S_0 = s)$ assuming the actions are selected according to π at time $1, \ldots, t-1$. When $t = 0$, $p_t^\pi(s'|a, s)$ becomes the point mass function $\mathbb{I}(s' = s)$ where $\mathbb{I}(\cdot)$ denotes the indicator function. When $t = 1$, p_t^π equals the transition function p defined in Condition (A1). We define the conditional discounted visitation probability function as $d^\pi(s'|a, s) = (1-\gamma)\sum_{t\geq 0}\gamma^t p_t^\pi(s'|a, s)$. Let $d^{\pi,\nu}(s') = \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\pi(a|s)d^\pi(s'|a, s)\nu(s)$, be the integrated discounted visitation probability function. Assume the actions are selected according to π and the initial state follows the distribution ν. Then $d^{\pi,\nu}$ corresponds to the probability mass function of a state S^* that is a mixture of the random variables $\{S_t\}_{t\geq 0}$ with the corresponding mixture weights $\{(1-\gamma)\gamma^t\}_{t\geq 0}$.

Finally, we introduce the discounted stationary probability ratio. For any π, define

$$\omega^\pi(a', s'; a, s) = \frac{(1-\gamma)\{\mathbb{I}(s' = s, a' = a) + \sum_{t\geq 1}\gamma^t\pi(a'|s')p_t^\pi(s'|a, s)\}}{p_\infty(a', s')}. \tag{4}$$

By definition, the denominator in (4) corresponds to the stationary distribution of (A_t, S_t). When the system follows π, the numerator corresponds to the probability mass function of the state-action pair (A^*, S^*) that is a mixture of $\{(A_t, S_t)\}_{t\geq 0}$ conditional on the event that $(A_0, S_0) = (a, s)$. We thus refer to ω^π as the conditional discounted stationary probability ratio. Similarly, define $\omega^{\pi,\nu}(a', s') = \sum_{a,s}\pi(a|s)\nu(s)\omega^\pi(a', s'; a, s)$ as the integrated discounted stationary probability ratio. We remark that these probability ratios play an important role in constructing debiased value function estimators for off-policy evaluation (Kallus and Uehara 2019; Shi et al. 2021).

2.3. Trust Region Policy Optimization

The following equation forms the basis of TRPO (Kakade and Langford 2002, Lemma 6.1),

$$(1-\gamma)\{\mathcal{V}(\pi_{\text{new}}) - \mathcal{V}(\pi_{\text{old}})\} = \sum_{a\in\mathcal{A},s\in\mathcal{S}} \pi_{\text{new}}(a|s)A^{\pi_{\text{old}}}(a, s)d^{\pi_{\text{new}},\nu}(s), \tag{5}$$

for any two policies π_{old} and π_{new}, where A^π and $d^{\pi,\nu}$ are defined in Section 2.2. Based on (5), for any given initial policy $\pi_{\text{old}} \in \Pi$, it is tempting to directly search $\pi_{\text{new}} \in \Pi$ that maximizes an estimates of the right-hand side (RHS) of (5). However, such a direct search method is computationally challenging, due to the complex dependency of $d^{\pi_{\text{new}},\nu}$ on π_{new}. In general, it is difficult to construct an estimator that has an explicit form of solution in terms of π_{new} for $d^{\pi_{\text{new}},\nu}$. The gradient of the corresponding estimator for (5) is extremely difficult to compute, and thus gradient-type methods are hard to apply.

To make the computation feasible and efficient, Schulman et al. (2015) considered approximating the RHS of (5) by

$$\eta_1(\pi_{\text{new}}, \pi_{\text{old}}) = \sum_{a\in\mathcal{A},s\in\mathcal{S}} \pi_{\text{new}}(a|s)A^{\pi_{\text{old}}}(a, s)d^{\pi_{\text{old}},\nu}(s). \tag{6}$$

Note that the discounted visitation probability in (6) now depends on π_{old}, rather than π_{new}. The quantity $(1-\gamma)\mathcal{V}(\pi_{\text{old}}) + \eta_1(\pi_{\text{new}}, \pi_{\text{old}})$ can be viewed as a first-order approximation of $(1-\gamma)\mathcal{V}(\pi_{\text{new}})$. To elaborate this more, note that $\eta_1(\pi_{\text{new}}, \pi_{\text{old}})$ can be rewritten as $\sum_{a\in\mathcal{A},s\in\mathcal{S}}\{\pi_{\text{new}}(a|s) - \pi_{\text{old}}(a|s)\}A^{\pi_{\text{old}}}(a, s)d^{\pi_{\text{old}},\nu}(s)$, by (3). It then follows from (5) that

$$\begin{aligned}(1-\gamma)\mathcal{V}(\pi_{\text{new}}) &= (1-\gamma)\mathcal{V}(\pi_{\text{old}}) \\ &+ \underbrace{\sum_{a\in\mathcal{A},s\in\mathcal{S}}\{\pi_{\text{new}}(a|s) - \pi_{\text{old}}(a|s)\}A^{\pi_{\text{old}}}(a, s)d^{\pi_{\text{old}},\nu}(s)}_{\eta_1(\pi_{\text{new}},\pi_{\text{old}})} \\ &+ \underbrace{\sum_{a\in\mathcal{A},s\in\mathcal{S}}\{\pi_{\text{new}}(a|s) - \pi_{\text{old}}(a|s)\}A^{\pi_{\text{old}}}(a, s)\{d^{\pi_{\text{new}},\nu}(s) - d^{\pi_{\text{old}},\nu}(s)\}}_{\eta_2(\pi_{\text{new}},\pi_{\text{old}})},\end{aligned}$$

where $\eta_2(\pi_{\text{new}}, \pi_{\text{old}})$ corresponds to a higher-order remainder term. To quantify this higher-order remainder term, for any two probability distributions μ_1, μ_2 on $\mathcal{A}$, we use $\mathcal{D}_{\text{TV}}(\mu_1, \mu_2)$ to denote the total variation distance $2^{-1}\sum_{a\in\mathcal{A}}|\mu_1(a) - \mu_2(a)|$. Let

$$\mathcal{D}_{\text{KL}}(\mu_1, \mu_2) = \sum_{a\in\mathcal{A}} \mu_1(a)\log\{\mu_1(a)/\mu_2(a)\}$$

denote the Kullback–Leibler (KL) divergence from μ_1 to μ_2. In Lemma A.1 (see Appendix A.1), we show that

$$\begin{aligned}|\eta_2(\pi_{\text{new}}, \pi_{\text{old}})| &\leq c^*\left[\mathbb{E}_{S^*\sim d^{\pi_{\text{old}},\nu}}\mathcal{D}_{\text{TV}}\left(\pi_{\text{old}}(\bullet|S^*), \pi_{\text{new}}(\bullet|S^*)\right)\right]^2 \\ &\leq c^*\mathbb{E}_{S^*\sim d^{\pi_{\text{old}},\nu}}\mathcal{D}_{\text{KL}}\left(\pi_{\text{old}}(\bullet|S^*), \pi_{\text{new}}(\bullet|S^*)\right),\end{aligned} \tag{7}$$

for some positive constant $c^* > 0$. The first inequality in (7) implies that $|\eta_2(\pi_{\text{new}}, \pi_{\text{old}})|$ is indeed a second-order term. Based on this observation, we consider a policy optimization procedure by maximizing a lower bound of (5), given by

$$\begin{aligned}\pi_{\text{new}} \in \operatorname{argmax}_{\pi\in\Pi}\ &[\eta_1(\pi, \pi_{\text{old}}) \\ &- c^*\mathbb{E}_{S^*\sim d^{\pi_{\text{old}},\nu}}\mathcal{D}_{\text{KL}}\left(\pi_{\text{old}}(\bullet|S^*), \pi(\bullet|S^*)\right)].\end{aligned} \tag{8}$$

Iteratively solving the above optimization yields a type of minorization-maximization (MM) algorithm (Hunter and Lange 2004) as we can see when $\pi = \pi_{\text{old}}$, the objective function in (8) becomes 0 and $(1-\gamma)\mathcal{V}(\pi)$ becomes $(1-\gamma)\mathcal{V}(\pi_{\text{old}})$. So one can guarantee that (5) is always nonnegative after optimization. Therefore, this type of algorithm can greatly reduce the computational cost by circumventing computing $d^{\pi_{\text{new}},\nu}$ and meanwhile monotonically improve the integrated value function. However, in practice, it may be hard to robustly choose the penalty coefficients c^* in (8). To resolve this issue,

one can consider iteratively solving the following equivalent optimization problem with a so-called trust region constraint:

$$\begin{aligned} &\pi_{\text{new}} \in \text{argmax}_{\pi\in\Pi}\ \eta_1(\pi,\pi_{\text{old}}) \\ &\text{subject to}\ \ \mathbb{E}_{S^*\sim d^{\pi_{\text{old}},\nu}} \mathcal{D}_{\text{KL}}\left(\pi_{\text{old}}(\bullet|S^*), \pi(\bullet|S^*)\right) \le \delta, \end{aligned} \tag{9}$$

for some constant $\delta > 0$. This yields the TRPO algorithm.

3. Value Enhanced Policy Optimization

In this section, we first present the motivation of our method. To implement TRPO, we need an estimate for $\eta_1(\pi,\pi_{\text{old}})$. In online settings, Schulman et al. (2015) proposed to simulate trajectories following the policy π_{old} to estimate $\eta_1(\pi,\pi_{\text{old}})$. In offline settings, it remains unknown how to effectively evaluate this quantity based on the observed data. By its definition, we note that η_1 depends on the nuisance functions $A^{\pi_{\text{old}}}$ and $d^{\pi_{\text{old}},\nu}$. A naive method is to first estimate these quantities (denote by $\widetilde{A}$ and $\widetilde{d}^{\nu}$) and then use the corresponding plug-in estimators $\widetilde{\eta}_1 = \mathbb{E}_{S^*\sim\widetilde{d}^{\nu}} \sum_{a\in\mathcal{A}} \pi(a|S^*)\widetilde{A}(a,S^*)$ to estimate $\eta_1(\pi,\pi_{\text{old}})$. However, such a procedure suffers from the following three main drawbacks:

1. Iteratively computing the optimization problem (9) can still be computationally expensive as the policy-dependent nuisance functions need to be updated at each iteration, especially when we do not have the closed-form expression for estimating these nuisance functions such as $\widetilde{d}^{\nu}$. Therefore, it may not be desirable to directly implement TRPO method.
2. When either $\widetilde{A}$ or $\widetilde{d}^{\nu}$ is not consistent, $\widetilde{\eta}_1$ might not be consistent. Consequently, there is no guarantee that the resulting new policy π_{new} can outperform π_{old}.
3. To ensure both $\widetilde{A}$ and $\widetilde{d}^{\nu}$ are both consistent, one might consider estimating these functions nonparametrically. Even when both of them are consistent, the plug-in estimator $\widetilde{\eta}_1$ might not be rate-optimal, that is, $(NT)^{-1/2}$, due to that the nonparametric estimators $\widetilde{A}$ and $\widetilde{d}^{\nu}$ usually converge much slower than $(NT)^{-1/2}$. Consequently, compared with π_{old}, the improvement by π_{new} may be marginal, resulting in a slow convergence rate to the optimal policy.

To address the first concern, we propose to first apply some existing state-of-the-art offline RL method to obtain a good initial policy. Several methods can be applied here, including the conservative Q-learning (CQL, Kumar et al. 2020), FQI, V-learning, among others. CQL and neural FQI use neural networks to index the policy class and V-learning considers a parametrized policy class indexed by a finite-dimensional vector. Note that these methods basically rely on the estimation of value or Q-functions. They are more computationally efficient than the iterative procedure described in (I). The value functions under initial policies obtained by these algorithms, if consistent, may converge at a slow rate as a tradeoff for fast computation. In the second step, we propose to solve (9) to improve their performances. This corresponds to a one-step update of the initial policy. One may also update this new policy for a few times to ensure the final estimated policy achieves a fast convergence rate. To remove the dependence between the initial policy and our policy optimization, we incorporate a data-splitting strategy, which is commonly seen in statistics and machine learning, for example, Chernozhukov et al. (2018), and Kallus and Uehara (2019). The detailed procedure is described in Section 3.2.

To address the second and the third concerns, we develop an efficient and robust estimating procedure for η_1, which is described in Section 3.1. Specifically, when the input policy π_{old} is consistent, we can guarantee that the output policy π_{new} by solving (9) achieves the desired "value enhancement" property. We call this set of methods "value enhanced policy optimization (VEPO)". An overview of our algorithm is given in Section 3.2, which integrates Q-learning, discounted stationary probability ratio estimation, transition dynamics estimation and policy search. We then discuss each component in the rest of the section.

3.1. An Efficient and Multiply Robust Estimator for η_1

For a given $\pi_{\text{old}} \in \Pi$, we first outline three potential approaches (see (i)–(iii)) to estimating $\eta_1(\pi,\pi_{\text{old}})$ from the observed data. Each of these methods requires some nuisance functions to be consistently estimated. We then present our proposal that combines these three methods to achieve efficient and triply robust estimation.

(i) Plug-in estimator: $\widetilde{\eta}_1^{(1)} = \mathbb{E}_{S^*\sim\widetilde{d}^{\nu}} \sum_{a\in\mathcal{A}} \pi(a|S^*)\widetilde{A}(a,S^*)$. This is the plug-in method discussed earlier. The validity of $\widetilde{\eta}_1^{(1)}$ requires the consistent estimation of $d^{\pi_{\text{old}},\nu}$ and $A^{\pi_{\text{old}}}$.

(ii) Importance sampling (IS) estimator I:

$$\begin{aligned} \widetilde{\eta}_1^{(2)} = \frac{1}{\sum_i T_i} \sum_{i=1}^{N} \sum_{t=0}^{T_i-1} \mathbb{E}_{S^*\sim\widetilde{d}^{\nu}} \sum_{a\in\mathcal{A}} \{\pi(a|S^*) \\ -\pi_{\text{old}}(a|S^*)\}\widetilde{\omega}(A_{i,t},S_{i,t};a,S^*)R_{i,t}, \end{aligned} \tag{10}$$

where $\widetilde{\omega}$ denotes some estimator for the conditional discounted probability ratio $\omega^{\pi_{\text{old}}}$. See (4) for a detailed definition. The validity of $\widetilde{\eta}_1^{(2)}$ requires consistent estimation of both $d^{\pi_{\text{old}},\nu}$ and $\omega^{\pi_{\text{old}}}$. Such an IS estimator is motivated by the work of Liu et al. (2018) on the off-policy value evaluation. A key observation is that, under (A2) and (A3), $Q^{\pi}(a,s)$ can be represented by

$$\begin{aligned} &\sum_{a',s'} \{\mathbb{I}(s'=s,a'=a) + \sum_{t\ge 1} \gamma^t \pi(a'|s') p_t^{\pi}(s'|a,s)\} r(s',a') \\ &\quad = \frac{1}{1-\gamma} \mathbb{E}\omega^{\pi}(A_t,S_t;a,s)R_t, \end{aligned}$$

for any t, s, and a. This yields the following IS estimator for $A^{\pi}(a,s)$:

$$\frac{1}{\sum_i T_i} \sum_{i=1}^{N} \sum_{t=0}^{T_i-1} \sum_{a'\in\mathcal{A}} \{\mathbb{I}(a'=a) - \pi(a'|s)\}\widetilde{\omega}(A_{i,t},S_{i,t};a',s)R_{i,t}.$$

Plugging in the above estimator for $A^{\pi_{\text{old}}}(a,s)$ and $\widetilde{d}^{\nu}$ for $d^{\pi_{\text{old}},\nu}$ yields (10).

(iii) IS estimator II:

$$\widetilde{\eta}_1^{(3)} = \frac{1}{\sum_i T_i} \sum_{i=1}^{N} \sum_{t=0}^{T_i-1} \sum_{a\in\mathcal{A}} \pi(a|S_{i,t})\widetilde{A}(a,S_{i,t})\widetilde{\omega}^{\nu}(A_{i,t},S_{i,t}), \tag{11}$$

where $\widetilde{\omega}^{\nu}$ denotes some estimator for the integrated probability ratio $\omega^{\pi_{\text{old}},\nu}$. The validity of $\widetilde{\eta}_1^{(3)}$ requires consistent estimation of $d^{\pi_{\text{old}},\nu}$ and the integrated probability ratio $\omega^{\pi_{\text{old}},\nu}$. To motivate

this estimator, we observe that the expectation $\mathbb{E}_{S\sim d^{\pi,\nu}} f(S)$ can be rewritten as $\mathbb{E} f(S_t)\omega^{\pi,\nu}(A_t,S_t)$, for any function f, policy π and decision point t. Consequently, we can represent η_1 by

$$\mathbb{E}\sum_{a\in\mathcal{A}}\pi(a|S_t)A^{\pi_{\text{old}}}(a,S_t)\omega^{\pi_{\text{old}},\nu}(A_t,S_t).$$

This yields the IS estimator in (11).

We note that each of the above estimator may be severely biased when the corresponding estimated nuisance functions fail to be consistent. Toward that end, we develop a multiply robust estimator by carefully combining the estimating strategies used in (i)–(iii). Meanwhile, the resulting estimator requires much weaker assumptions to achieve consistency. Let o be a shorthand for a data tuple (s,a,r,s'). The key to constructing our estimator is the following estimating function,

$$\begin{aligned}&\psi(o;\pi,\pi_{\text{old}},\widetilde{V},\widetilde{A},\widetilde{\omega},\widetilde{d})\\&=\psi_1(\pi,\pi_{\text{old}},\widetilde{A},\widetilde{d})\\&\quad+\psi_2(o;\pi,\pi_{\text{old}},\widetilde{V},\widetilde{A},\widetilde{\omega},\widetilde{d})+\psi_3(o;\pi,\pi_{\text{old}},\widetilde{A},\widetilde{\omega},\widetilde{d}),\end{aligned}$$

for some given nuisance functions $\widetilde{V}$, $\widetilde{A}$, $\widetilde{\omega}$, and $\widetilde{d}$, where,

$$\begin{aligned}&\psi_1(\pi,\pi_{\text{old}},\widetilde{A},\widetilde{d})=\mathbb{E}_{S^*\sim\widetilde{d}^{\nu}}\sum_{a\in\mathcal{A}}\pi(a|S^*)\widetilde{A}(a,S^*),\\&\psi_2(o;\pi,\pi_{\text{old}},\widetilde{V},\widetilde{A},\widetilde{\omega},\widetilde{d})\\&=\frac{1}{1-\gamma}\mathbb{E}_{S^*\sim\widetilde{d}^{\nu}}\sum_{a^*\in\mathcal{A}}\{\pi(a^*|S^*)-\pi_{\text{old}}(a^*|S^*)\}\widetilde{\omega}(a,s;a^*,S^*)\\&\quad\times\{r+\gamma\widetilde{V}(s')-\widetilde{V}(s)-\widetilde{A}(a,s)\}\psi_3(o;\pi,\pi_{\text{old}},\widetilde{A},\widetilde{\omega},\widetilde{d})\\&=\sum_{a^*\in\mathcal{A}}\frac{\widetilde{\omega}^{\nu}(a,s)}{1-\gamma}\Big[\gamma\mathbb{E}_{a'\sim\pi_{\text{old}}(\bullet|s')S^*\sim\widetilde{d}(\bullet|a',s')}\widetilde{A}(a^*,S^*)\pi(a^*|S^*)\\&\quad-\mathbb{E}_{S^*\sim\widetilde{d}(\bullet|a,s)}\widetilde{A}(a^*,S^*)\pi(a^*|S^*)+(1-\gamma)\pi(a^*|s)\widetilde{A}(a^*,s)\Big],\end{aligned}$$

where the nuisance functions $\widetilde{d}^{\nu}$ and $\widetilde{\omega}^{\nu}$ are determined by $\widetilde{d}$ and $\widetilde{\omega}$, given by $\widetilde{d}^{\nu}(\bullet)=\sum_{a,s}\pi_{\text{old}}(a|s)\nu(s)\widetilde{d}(\bullet|a,s)$ and $\widetilde{\omega}^{\nu}(\bullet,\bullet)=\sum_{a,s}\pi_{\text{old}}(a|s)\nu(s)\widetilde{\omega}(\bullet,\bullet;a,s)$.

By definition, ψ consists of three terms. The first term ψ_1 is essentially the plug-in estimator that depends only on $\widetilde{A}$ and $\widetilde{d}$. The second and third terms, that is, ψ_2 and ψ_3, are the augmentation terms. Let $O_t=(S_t,A_t,R_t,S_{t+1})$ for any t, we have $\mathbb{E}\psi_2(O_t;\pi,\pi_{\text{old}},\widetilde{V},\widetilde{A},\widetilde{\omega},\widetilde{d})=0$ when $\widetilde{A}=A^{\pi_{\text{old}}}$, $\widetilde{V}=V^{\pi_{\text{old}}}$ and $\mathbb{E}\psi_3(O_t;\pi,\pi_{\text{old}},\widetilde{A},\widetilde{\omega},\widetilde{d})=0$ when $\widetilde{d}=d^{\pi_{\text{old}}}$. See Appendix A.2 for details. The purpose of adding these two terms is to offer an additional protection against the potential bias of ψ_1 resulting from the biases of $\widetilde{A}$ and $\widetilde{d}$. Therefore, we have the following proposition.

Proposition 1. Suppose $\sum_a\pi_{\text{old}}(a|s)\widetilde{A}(a,s)=0$ for any s. Then $\psi(O_t;\pi,\pi_{\text{old}},\widetilde{V},\widetilde{A},\widetilde{\omega},\widetilde{d})$ is unbiased to η_1 as long as one of the following three assumptions are satisfied: (B1) $\widetilde{A}=A^{\pi_{\text{old}}}$, $\widetilde{V}=V^{\pi_{\text{old}}}$ and $\widetilde{d}=d^{\pi_{\text{old}}}$; (B2) $\widetilde{\omega}=\omega^{\pi_{\text{old}}}$ and $\widetilde{d}=d^{\pi_{\text{old}}}$; (B3) $\widetilde{A}=A^{\pi_{\text{old}}}$ and $\widetilde{\omega}=\omega^{\pi_{\text{old}}}$.

The condition $\sum_a\pi_{\text{old}}(a|s)\widetilde{A}(a,s)$ is automatically satisfied if we set $\widetilde{A}(a,s)=A^*(a,s)-\sum_a\pi_{\text{old}}(a|s)\widetilde{A}^*(a,s)=0$ for any initial advantage estimator $\widetilde{A}^*$. We remark that if $Q^{\pi_{\text{old}}}$ is correctly specified, so do $A^{\pi_{\text{old}}}$ and $V^{\pi_{\text{old}}}$. Based on this estimating function, a triply-robust estimator for η_1 is given by

$$\frac{1}{\sum_i T_i}\sum_{i=1}^{N}\sum_{t=0}^{T_i-1}\psi(O_{i,t};\pi,\pi_{\text{old}},\widetilde{V},\widetilde{A},\widetilde{\omega},\widetilde{d}),\tag{12}$$

where $O_{i,t}=(S_{i,t},A_{i,t},R_{i,t},S_{i,t+1})$. It remains to specify the estimation of nuisance functions. We present the details in the next section. In Section 4.2, we show the resulting estimator is efficient.

3.2. The Complete Algorithm

Our main idea is to construct an efficient and robust estimator for η_1 to improve the performance of an initial policy π_{old}. To achieve this goal, we need to estimate four key nuisance functions: (a) An initial policy π_{old}; (b) The value and advantage function $V^{\pi_{\text{old}}}$ and $A^{\pi_{\text{old}}}$; (c) The conditional discounted stationary probability ratio $\omega^{\pi_{\text{old}}}$; (d) The conditional discounted visitation probability function $d^{\pi_{\text{old}}}$.

Correspondingly, our estimating procedure involves four key steps, described in Sections 3.2.1–3.2.4, respectively. In addition to these four main estimating components, we also propose to couple the estimator in (12) with a data-splitting and cross-fitting strategy. Specifically, without loss of generality, we randomly divide the indices of all trajectories $\{1,2,\ldots,N\}$ into $\mathbb{L}$ subsets $\cup_{\ell=1}^{\mathbb{L}}\{O_{i,t}\}_{i\in\mathcal{I}_\ell,0\le t<T_i}$ with equal size, where $\mathcal{I}_\ell$ denotes the indices of trajectories contained in the ℓth data subset. We next apply the learning components in (a)-(d) to the data subsets in $\mathcal{I}_\ell^c=\{1,\ldots,N\}\backslash\mathcal{I}_\ell$ for $\ell=1,\ldots,\mathbb{L}$ and construct the estimator $\widehat{\eta}_1$ via cross-fitting. Cross-fitting essentially guarantees that the dataset used to learn (a)–(d) is independent of the dataset used to construct $\widehat{\eta}_1$. This allows us to avoid imposing Donsker-typed conditions, which limit the growth rate of the VC dimension of the estimators for (a)–(d) (Chernozhukov et al. 2018), to achieve desirable properties of our procedure. Then we propose to search π_{new} that maximizes $\widehat{\eta}_1$ subject to the trust region constraint in (9) with $d^{\pi_{\text{old}},\nu}$ replaced by its corresponding estimator. This corresponds to a one-step update of the initial policy. After computing π_{new}, we can repeat the above procedures a few times to guarantee the final estimated policy achieves a fast convergence rates. See Section 3.2.5 for details. A pseudocode summarizing our approach is given in Algorithm 1.

We remark that it is not necessary to develop a robust and efficient estimating procedure for the integrated KL divergence in the trust region constraint, due to the fact that it corresponds to a higher-order remainder term for the value difference (see Lemma A.1 in Appendix A.1). Our proposal works as long as $d^{\pi_{\text{old}},\nu}$ and its estimator have the common support. This condition is automatically satisfied when the reference distribution ν is uniformly bounded away from zero on $\mathcal{S}$.

3.2.1. Step 1: Initial Policy Optimization

First, to initial our VEPO algorithm, we need to estimate an initial policy denoted by π_{old}. We propose to apply some existing state-of-the-art offline RL algorithm on the data subset $\mathcal{I}_\ell^c$ and obtain resulting estimated policy denoted by $\hat{\pi}_{\text{old}}^{(\ell)}$ for $\ell=1,\ldots,\mathbb{L}$. For the ease of presentation, we often write $\hat{\pi}_{\text{old}}^{(\ell)}$ as

Algorithm 1 Value enhanced policy optimization

Input: A policy class Π and the observed data.

Output: An updated policy π_{new}.

Step 0. Randomly split the trajectories into $\mathbb{L}$ disjoint subsets, $\cup_{\ell=1}^{\mathbb{L}}\mathcal{I}_\ell$. Let $\mathcal{I}_\ell^c = \{1,\ldots,N\} - \mathcal{I}_\ell$, for $\ell = 1,\ldots,\mathbb{L}$.

1. For $\ell = 1,\ldots,\mathbb{L}$: apply some existing state of art offline RL algorithm to obtain the input initial policy π_{old} using the data subset $\mathcal{I}_\ell^c$. Denote the resulting policy as $\pi_{\text{old}}^{(\ell)}$.
2. For $\ell = 1,\ldots,\mathbb{L}$:
   (a) Apply fitted-Q evaluation (see (14)) to estimate the Q-function $Q^{\pi_{\text{old}}^{(\ell)}}$, based on the data subset in $\mathcal{I}_\ell^c$. Denote the resulting estimator by $\widehat{Q}^{(\ell)}$.
   (b) Set $\widehat{V}^{(\ell)}(s) = \sum_a \pi_{\text{old}}^{(\ell)}(a|s)\widehat{Q}^{(\ell)}(a,s)$ for any s and $\widehat{A}^{(\ell)}(a,s) = \widehat{Q}^{(\ell)} - \widehat{V}^{(\ell)}(s)$ for any a and s.
3. For $\ell = 1,2,\ldots,\mathbb{L}$, apply the method detailed in Section 3.2.3 to learn a conditional probability ratio $\widehat{\omega}^{(\ell)}$, based on the data subset $\mathcal{I}_\ell^c$.
4. For $\ell = 1,2,\ldots,\mathbb{L}$, apply machine learning methods to approximate the conditional distribution of S_{t+1} given A_t and S_t, using the data subset $\mathcal{I}_\ell^c$, which can be used to generate the pseudo sample. The pseudo sample can then be used to approximate the distribution function $d^{\pi_{\text{old}}^{(\ell)}}$ and $d^{\pi_{\text{old}}^{(\ell)},\nu}$ (see Algorithms 2 and 3).
5. Construct the estimator for η_1 and update the corresponding policy:
   (a) Apply cross-fitting to construct the value difference estimator $\widehat{\eta}_1$ (see (19)).
   (b) Use the estimated dynamic to construct the trust region constraint (see (20)).
   (c) Search $\pi_{\text{new}} \in \Pi$ that maximizes $\widehat{\eta}_1$ subject to (20).
6. Set all $\pi_{\text{old}}^{(\ell)}$ to π_{new} for $\ell = 1,\ldots,\mathbb{L}$ and repeat Steps 2–5 a few times.

$\pi_{\text{old}}^{(\ell)}$ when there is no confusion. Specifically, in our numerical studies, we implement three offline RL algorithms to obtain our initial policies. The first one is FQI using the idea of value iteration with function approximation (Sutton and Barto 2018). It relies on the optimal Bellman equation (Bertsekas and Tsitsiklis 1996). The second one is V-learning proposed by Luckett et al. (2020), which considered policy iteration with function approximation. The last one is CQL by Kumar et al. (2020), which proposed to learn a lower bound of Q-function during the policy iteration procedure. The last method is driven by the overestimation of the value function due to the distributional mismatch between the behavior policy in the batch dataset and the learned policy. We remark that any valid offline RL method can be employed here, as long as assumptions in Theorem 2 are satisfied. See details in Section 4.1.

3.2.2. Step 2: Q-learning

Second, to estimate nuisance functions in (b), we employ a Q-learning type algorithm to learn the Q-function $Q^{\pi_{\text{old}}^{(\ell)}}$, based on the data subset in $\mathcal{I}_\ell^c$. Denote the corresponding estimator by $\widehat{Q}^{(\ell)}$. We then construct the corresponding estimators for the value and advantage function by $\widehat{V}^{(\ell)}(s) = \sum_a \pi_{\text{old}}^{(\ell)}(a|s)\widehat{Q}^{(\ell)}(a,s)$ and $\widehat{A}^{(\ell)}(a,s) = \widehat{Q}^{(\ell)}(a,s) - \widehat{V}^{(\ell)}(s)$, for any s and a, based on the relation that $V^{\pi_{\text{old}}^{(\ell)}}(s) = \sum_a \pi_{\text{old}}^{(\ell)}(a|s)Q^{\pi_{\text{old}}^{(\ell)}}(a,s)$, $A^{\pi_{\text{old}}^{(\ell)}}(a,s) = Q^{\pi_{\text{old}}^{(\ell)}}(a,s) - V^{\pi_{\text{old}}^{(\ell)}}(s)$. Consequently, the requirement $\sum_a \pi_{\text{old}}^{(\ell)}(a|s)\widehat{A}^{(\ell)}(a,s) = 0$ for the estimated advantage function is automatically satisfied.

Several algorithms can be used here to estimate the Q-function $Q^{\pi_{\text{old}}^{(\ell)}}$. Here, we adopt the fitted Q-evaluation evaluation (FQE) method proposed by Le, Voloshin, and Yue (2019). The following Bellman's equation forms the basis of all Q-learning type algorithms: for $t \geq 0$,

$$Q^{\pi_{\text{old}}^{(\ell)}}(A_t,S_t) = \mathbb{E}\left\{R_t + \gamma \sum_a \pi_{\text{old}}^{(\ell)}(a|S_{t+1})Q^{\pi_{\text{old}}^{(\ell)}}(a,S_{t+1}) \,\middle|\, A_t,S_t\right\}. \tag{13}$$

Based on this identity, we estimate $Q^{\pi_{\text{old}}^{(\ell)}}$ by iteratively computing

$$\widehat{Q}_k^{(\ell)} = \text{argmin}_{Q_k} \sum_{i,t} \left\{R_{i,t} + \sum_a \pi_{\text{old}}^{(\ell)}(a|S_{i,t+1})\widehat{Q}_{k-1}^{(\ell)}(a,S_{i,t+1}) - Q_k(A_{i,t},S_{i,t})\right\}^2, \tag{14}$$

for $k = 1,2,\ldots$ with any initial $\widehat{Q}_0^{(\ell)}$. Several supervised learning methods can be incorporated here, since (14) is essentially a regression problem. In our implementation, we employ deep learning (LeCun, Bengio, and Hinton 2015) to compute $\widehat{Q}_k^{(\ell)}$ during each iteration.

3.2.3. Step 3: Discounted Stationary Probability Ratio Estimation

We adopt the algorithm developed by Shi et al. (2021) to estimate (c). The procedure is motivated by the following observation: For any two pairs (i,t) and (i',t') such that $O_{i,t}$ and $O_{i',t'}$ are independent, we have for any function f such that $\mathbb{E}\Delta(\omega^\pi,f,\pi;i,t,i',t') = 0$, where

$$\begin{aligned}&\Delta(\omega^\pi,f,\pi;i,t,i',t') = \omega^\pi(S_{i',t'},A_{i',t'};S_{i,t},A_{i,t})\\ &\left\{\gamma\sum_a \pi(a|S_{i',t'+1})f(S_{i',t'+1},a;S_{i,t},A_{i,t}) - f(S_{i',t'},A_{i',t'};S_{i,t},A_{i,t})\right\}\\ &+(1-\gamma)f(S_{i,t},A_{i,t},S_{i,t},A_{i,t}).\end{aligned} \tag{15}$$

Conversely, for any ω that satisfies $\mathbb{E}\Delta(\omega,f,\pi;i,t,i',t') = 0$ for any f, we have $\omega = \omega^\pi$. See (5) of Shi et al. (2021) for details.

For each function f, an estimating equation for $\omega^{\pi_{\text{old}}^{(\ell)}}$ can be constructed based on (15). Similar to the proposal in Liu et al. (2018), f can be treated as a discriminator to construct the following minimax loss function

$$\text{argmin}_{\omega\in\Omega}\sup_{f\in\mathcal{F}}\left|\mathbb{E}\Delta(\omega^{\pi_{\text{old}}^{(\ell)}},f,\pi_{\text{old}}^{(\ell)};i,t,i',t')\right|^2, \tag{16}$$

for some function classes Ω and $\mathcal{F}$. To simplify the calculation, $\mathcal{F}$ is set to a unit ball of a reproducing kernel Hilbert space. This yields a closed-form expression for the inner maximization problem in (16). The expectation in (16) is then approximated by the empirical distributions of the data subset in $\mathcal{I}_\ell^c$. The parameters involved in ω are updated by the stochastic gradient descent algorithm. To save space, we present the details in Section B.1 of the supplementary material.

3.2.4. Step 4: Estimation of the Underlying Dynamics

The conditional discounted visitation probability $d^{\pi_{\text{old}}^{(\ell)}}$ in (d) is extremely difficult to estimate when the state space is high-dimensional, as it corresponds to a mixture distribution of state variables at different decision points. A key observation is that, $d^{\pi_{\text{old}}^{(\ell)}}$ is completely determined by the system dynamics. As long as the transition kernel p can be consistently estimated, $d^{\pi_{\text{old}}^{(\ell)}}$ can be well-approximated.

To compute $d^{\pi_{\text{old}}^{(\ell)}}$, in continuous state space, we propose to use a Gaussian probabilistic model to estimate the transition kernel p with the use of machine learning models to approximate the mean and covariance matrix. In particular, we assume the next state S_{t+1} given the current action-state (S_t, A_t) follows a multi-variate Gaussian distribution $\mathcal{N}(\mu(S_t, A_t), \Sigma(S_t, A_t))$, where μ and Σ are the corresponding mean and covariance matrix functions, respectively. Notice that estimating μ is essentially a regression problem, and there are many supervised learning methods available. In our implementation, we use deep neural networks to compute $\widehat{\mu}^{(\ell)}$, via

$$\widehat{\mu}_j^{(\ell)} = \arg\min_{\mu_j} \sum_{i\in\mathcal{I}_\ell^c} \sum_{t=0}^{T_i-1} [S_{i,t+1,j} - \mu_j(S_{i,t}, A_{i,t})]^2,$$

where $\widehat{\mu}_j^{(\ell)}$ and $S_{i,t+1,j}$ denote the jth element of $\widehat{\mu}^{(\ell)}$ and $S_{i,t+1}$, respectively. Next, let $\varepsilon_{i,t,j}$ denote the residual $S_{i,t+1,j} - \widehat{\mu}_j^{(\ell)}(S_{i,t}, A_{i,t})$. We employ deep learning again to compute $\widehat{\Sigma}^{(\ell)}$, via

$$\widehat{\Sigma}_{j_1,j_2}^{(\ell)} = \arg\min_{\Sigma_{j_1,j_2}} \sum_{i\in\mathcal{I}_\ell^c} \sum_{t=0}^{T_i-1} [\varepsilon_{i,t,j_1}\varepsilon_{i,t,j_2} - \Sigma_{j_1,j_2}(S_{i,t}, A_{i,t})]^2,$$

where $\widehat{\Sigma}_{j_1,j_2}^{(\ell)}$ denotes the (j_1, j_2)th entry of $\widehat{\Sigma}^{(\ell)}$, which is the final estimator of $\Sigma^{(\ell)}$.

To approximate the conditional distribution $d^{\pi_{\text{old}}^{(\ell)}}(\bullet|a, s)$ for any a and s, we can employ the Monte Carlo method and generate a sequence of pseudo samples $\{\widetilde{S}_{t'}^{(m)}\}_{1\le m\le M, 0\le t\le T'}$ based on $\widehat{\mu}^{(\ell)}$ and $\widehat{\Sigma}^{(\ell)}$. We illustrate the details in Algorithm 2. For any function f, the integral $\mathbb{E}_{S^*\sim d^{\pi_{\text{old}}^{(\ell)}}(\bullet|a,s)} f(S^*)$ can then be approximated by

$$(1-\gamma)M^{-1}\sum_{m=1}^{M}\sum_{t'=0}^{T'}\gamma^t f(\widetilde{S}_{t'}^{(m)}), \tag{17}$$

which we denote by $\mathbb{E}_{S^*\sim\widehat{d}^{(\ell)}(\bullet|a,s)} f(S^*)$ where $\widehat{d}^{(\ell)}$ denotes the estimated probability density/mass function. We use the aggregated squared total variation distance

$$\mathbb{E}_{(S,A)\sim p_\infty} \mathcal{D}_{\text{TV}}^2(\widehat{d}^{(\ell)}(\bullet|A, S), d^{\pi_{\text{old}}^{(\ell)}}(\bullet|A, S)) \tag{18}$$

Algorithm 2 Generate pseudo samples to approximate $d^{\pi_{\text{old}}^{(\ell)}}(\bullet|s, a)$

Input: Estimators $(\widehat{\mu}^{(\ell)}, \widehat{\Sigma}^{(\ell)})$ computed via supervised learning.

for $m = 1$ to M: **do**

(a)Set $\widetilde{S}_0^{(m)} = s$ and $\widetilde{A}_0^{(m)} = a$;

(b)For $t' = 1$ to T', generate $\widetilde{S}_{t'}^{(m)}$ by $\mathcal{N}(\widehat{\mu}^{(\ell)}(\widetilde{A}_{t'-1}^{(m)}, \widetilde{S}_{t'-1}^{(m)}), \widehat{\Sigma}^{(\ell)}(\widetilde{A}_{t'-1}^{(m)}, \widetilde{S}_{t'-1}^{(m)}))$ and randomly sample $\widetilde{A}_{t'}^{(m)}$ from $\pi_{\text{old}}^{(\ell)}(\bullet|\widetilde{S}_{t'}^{(m)})$.

Output $\{\widetilde{S}_{t'}^{(m)}\}_{1\le m\le M, 0\le t\le T'}$.

Algorithm 3 Generate pseudo samples to approximate $d^{\pi_{\text{old}}^{(\ell)}, \nu}$

Input: Estimators $(\widehat{\mu}^{(\ell)}, \widehat{\Sigma}^{(\ell)})$ computed via supervised learning.

for $m = 1$ to M: **do**

(a)Sample $\widetilde{S}_0^{(m),\nu}$ from ν;

(b)For $t' = 0$ to $T' - 1$, randomly sample $\widetilde{A}_{t'}^{(m),\nu}$ from $\pi_{\text{old}}^{(\ell)}(\bullet|\widetilde{S}_{t'}^{(m),\nu})$ and generate generate $\widetilde{S}_{t'+1}^{(m),\nu}$ by $\mathcal{N}(\widehat{\mu}^{(\ell)}(\widetilde{A}_{t'}^{(m)}, \widetilde{S}_{t'}^{(m),\nu}), \widehat{\Sigma}^{(\ell)}(\widetilde{A}_{t'}^{(m)}, \widetilde{S}_{t'}^{(m),\nu}))$.

Output $\{\widetilde{S}_{t'}^{(m),\nu}\}_{1\le m\le M, 0\le t\le T'}$.

to measure its goodness of fit. See Condition (C3) in Section 4 for details. In Section B.2 of the supplementary material, we show that when the conditional Gaussian model is correctly specified, the minimum eigenvalue of $\Sigma(a, s)$ is uniformly bounded away from zero for any a, s, and $\widehat{\Sigma}(a, s)$ is positive definite for any a, s, (18) is upper bounded by

$$3\gamma^{2T'} + \frac{3}{M} + O(1)\mathbb{E}_{(A^*,S^*)\sim p_\infty}[\|\mu(S^*, A^*) - \widehat{\mu}^{(\ell)}(S^*, A^*)\|_2 + \|\Sigma(S^*, A^*) - \widehat{\Sigma}^{(\ell)}(S^*, A^*)\|_F]^2,$$

where $O(1)$ denotes some positive constant. As such $\widehat{d}^{(\ell)}$ is consistent as long as $T', M \to \infty$ and that the estimated conditional mean and covariance functions are consistent. We also remark that the conditional Gaussian model is widely used in the RL literature for learning the transition function (see e.g., Yu et al. 2020). Alternatively, a conditional Gaussian mixture model can be employed to mitigate model misspecification (Bishop 1994).

3.2.5. Step 5: Policy Optimization

After obtaining the four key components, we next discuss the procedure to compute the new policy π_{new}. We propose to construct the estimator $\widehat{\eta}_1$ via cross-fitting. Specifically, we estimate it by

$$\widehat{\eta}_1(\pi) = \frac{1}{\sum_i T_i}\sum_{\ell=1}^{\mathbb{L}} \times \left\{\sum_{i\in\mathcal{I}_\ell}\sum_{t=0}^{T_i-1} \psi(O_{i,t}; \pi, \pi_{\text{old}}^{(\ell)}, \widehat{V}^{(\ell)}, \widehat{A}^{(\ell)}, \widehat{\omega}^{(\ell)}, \widehat{d}^{(\ell)})\right\}. \tag{19}$$

Note that the nuisance functions $\pi_{\text{old}}^{(\ell)}$, $\widehat{A}^{(\ell)}$, $\widehat{V}^{(\ell)}$, $\widehat{\omega}^{(\ell)}$, and $\widehat{d}^{(\ell)}$ are computed based on the data subset in $\mathcal{I}_\ell^c$ and are independent of the observations in $\mathcal{I}_\ell$ that are used to construct the estimating function ψ. Theoretical properties of this estimator are studied in Section 4.2.

We then propose to learn π_{new} by solving the following constrained optimization,

$$\pi_{\text{new}} \in \text{argmax}_{\pi\in\Pi}\widehat{\eta}_1(\pi),$$
$$\text{subject to } \frac{1}{\mathbb{L}}\sum_{\ell=1}^{\mathbb{L}}\mathbb{E}_{S^*\sim\widehat{d}^{(\ell),v}}\mathcal{D}_{\text{KL}}\left(\pi_{\text{old}}^{(\ell)}(\bullet|S^*),\pi(\bullet|S^*)\right)\le\delta, \tag{20}$$

where $\widehat{d}^{(\ell),v}$ denotes the distribution of the pseudo samples $\{\widetilde{S}_{t'}^{(m),v}\}_{1\le m\le M,0\le t\le T'}$ generated according to Algorithm 3.

We remark that when the initial policy $\pi_{\text{old}}^{(\ell)}$ is set to the behavior policy b, the constraint in (20) then requires the learned policy close to the behavior one in the batch dataset, which is commonly used in the recently developed RL algorithms (e.g., Wu et al. 2019). As pointed out by Levine et al. (2020), one of the fundamental challenges of offline RL is the out of distribution due to the mismatch between behavior policy and the target policy. This out of distribution issue will result in an overestimation for the value function, therefore, deteriorating the performance of policy learning. Restricting the learned policies to stay close to the behavior one can potentially relieve this limitation. In practice, we can repeat the constraint optimization in (20) several times by setting all the initial policies $\pi_{\text{old}}^{(\ell)}$ for $\ell = 1,\ldots,\mathbb{L}$ to π_{new} obtained from the previous iteration. This guarantees that the final estimated policy achieves a fast convergence rate. Finally, we remark that our method is not overly complicated compared to the existing state-of-the-art RL algorithms and indeed quite flexible. Although we require to learn a number of components and use deep learning models in our numerical experiments below, these components can be alternatively estimated via much simpler methods (e.g., parametric models, sieve methods or kernels). In this case, our proposed algorithm becomes more accessible.

4. Theory

In this section, we systematically study the theoretical properties of our algorithm. In Section 4.1, we establish the properties of our estimated optimal policy. In Section 4.2, we show the proposed first-order value difference estimator $\widehat{\eta}_1$ is efficient. To simplify the theoretical analysis, we assume $T_1 = \cdots = T_N = T$. All the asymptotic results are derived when either the number of trajectories N, or the number of decision points T, diverges to infinity. Results of this type provide useful theoretical guarantees for a variety of applications in reinforcement learning. We refer to theories of this type as bidirectional theories. We also allow the state-action space, the transition matrix p, the reward function r and policy class Π to depend on N and T. Consequently, the optimal policy, the Q function and the discounted visitation probability are allowed to vary with N or T.

4.1. Properties of the Estimated Optimal Policy

We first show in Theorem 1 that the value difference between the new and the old policy is $O(\sqrt{\delta})$ where δ corresponds to the threshold in the trust region constraint (20). Consequently, by setting $\delta \to 0$, we can guarantee that the new policy is asymptotically no worse than the old one on average.

Theorem 1. $|\mathcal{V}(\pi_{\text{new}}) - \mathbb{L}^{-1}\sum_{\ell=1}^{\mathbb{L}}\mathcal{V}(\pi_{\text{old}}^{(\ell)})| \le O(1)\sqrt{\delta}$ where $O(1)$ denotes some positive constant.

We remark that Theorem 1 does not require any conditions on the estimated nuisance functions $\widehat{Q}^{(\ell)}$, $\widehat{\omega}^{(\ell)}$, and $\widehat{d}^{(\ell)}$. Nor does it require $\pi_{\text{old}}^{(\ell)}$ to converge to an optimal policy π^{opt}. In addition, Theorem 1 holds deterministically even if the policy $\pi_{\text{old}}^{(\ell)}$ is data-dependent. See Appendix A.1 for more details.

Note that $N \times T$ corresponds to the total number of decision points. We next consider the scenario where the input policy $\pi_{\text{old}}^{(\ell)}$ is close to π^{opt} in the sense that $\mathcal{V}(\pi_{\text{old}}^{(\ell)}) = \mathcal{V}(\pi^{\text{opt}}) + O\{(NT)^{-\kappa_0}\}$ for some constant $\kappa_0 > 0$. This implies that $\pi_{\text{old}}^{(\ell)}$ is consistent to π^{opt} as either N or T diverges to infinity. We further assume $\mathcal{V}(\pi^*) = \mathcal{V}(\pi^{opt}) + o(1)$, as $NT \to \infty$. Recall that π^* is defined as the optimal in-class policy that maximizes the value among Π. In other words, the value under the optimal in-class policy approaches to the optimal value function as the sample size increases (because the size of policy class also increases). To simplify the theoretical analysis, we assume $\pi^{\text{opt}} \in \Pi$ such that $\pi^* = \pi^{\text{opt}}$. Meanwhile, our theories are equally applied to settings where $\pi^{\text{opt}} \notin \Pi$ but the value difference $\mathcal{V}(\pi^*) - \mathcal{V}(\pi^{\text{opt}})$ converges at a sufficiently fast rate. This assumption is reasonable in practice when we either have domain knowledge on the parametric form of π^{opt} or use function classes with the universal approximation capabilities (e.g., neural networks) to parameterize Π. To establish the value enhancement property, we need the following set of conditions.

(C1) Suppose $\mathbb{E}_{(A,S)\sim p_\infty}|\widehat{Q}^{(\ell)}(A,S) - Q^{\pi_{\text{old}}^{(\ell)}}(A,S)|^2 = O_p\{(NT)^{-2\kappa_1}\}$ for some constant $\kappa_1 \ge 0$. In addition, $\widehat{Q}^{(\ell)}$ is uniformly bounded almost surely.

(C2) Suppose $\mathbb{E}_{(A,S),(\widetilde{A},\widetilde{S})\sim p_\infty}|\widehat{\omega}^{(\ell)}(\widetilde{A},\widetilde{S};A,S) - \omega^{\pi_{\text{old}}^{(\ell)}}(\widetilde{A},\widetilde{S};A,S)| = O_p\{(NT)^{-2\kappa_2}\}$ for some constant $\kappa_2 \ge 0$, where $(\widetilde{A},\widetilde{S})$ and (A,S) denote two independent state-action pairs generated according to p_∞. In addition, $\widehat{\omega}^{(\ell)}$ is uniformly bounded almost surely.

(C3) Suppose $\mathbb{E}_{(A,S)\sim p_\infty}|\mathcal{D}_{\text{TV}}(d^{\pi_{\text{old}}^{(\ell)}}(\bullet|A,S),\widehat{d}^{(\ell)}(\bullet|A,S))|^2 = O_p\{(NT)^{-2\kappa_3}\}$ for some constant $\kappa_3 \ge 0$.

(C4) Suppose Π corresponds to certain VC type function class (Chernozhukov, Chetverikov, and Kato 2014) with VC indices upper bounded by $O\{(NT)^{\kappa_4}\}$ for some constant $0 \le \kappa_4 < \frac{\alpha}{\alpha+1}$, where α is defined below.

(C5) The optimal policy is unique. In addition, there exist some positive constants $\alpha, \bar{c}, \bar{\epsilon}$ such that $\Pr(-\epsilon \le A^{\pi^{\text{opt}}}(a,S^*) < 0) \le \bar{c}\epsilon^\alpha$ for any $a \in \mathcal{A}$ and $0 < \epsilon \le \bar{\epsilon}$, where the random variable S^* is distributed according to $d^{\pi^{\text{opt}},v}$.

(C6) The process $\{(S_t,A_t,R_t)\}_{t\ge0}$ is exponentially β-mixing.

Conditions (C1)–(C3) characterize the theoretical requirements on the learners in (a)–(c), respectively. In particular, (C1)–(C2) require the squared prediction losses of the estimated

Q-function and the conditional probability ratio to satisfy certain convergence rates, whereas Condition (C3) assumes the squared total variation norm between the transition function and its estimator to satisfy a certain convergence rate. If some parametric models are imposed to learn $Q^{\pi_{\text{old}}}$, $\omega^{\pi_{\text{old}}}$ and the transition matrix p, we have $\kappa_1 = \kappa_2 = \kappa_3 = 1/2$. In our setup, we only require $\kappa_{i_1} + \kappa_{i_2} > 1/(2+2\alpha)$ for any disjoint $i_1, i_2 \in \{1,2,3\}$. See the statement of Theorem 2. This condition holds when $\min_{i\in\{1,2,3\}} \kappa_i > 1/(4+4\alpha)$ and thus is achievable for many nonparametric estimators. It is also strictly weaker than those imposed in the recent literature that require the nuisance function to converge at a rate faster than $(NT)^{-1/4}$ for off-policy value evaluation (e.g., Kallus and Uehara 2019). For example, when the kernel smoother (Feng et al. 2020), sieve method (Shi et al. 2020b; Chen and Qi 2022) or deep neural networks (Fan et al. 2020b) are used to approximate the Q-function it can be shown that under some technical conditions, (C2) holds with $\kappa_1 = \beta_1/(2\beta_1 + d_S)$ and $\beta_1 > d_S/2$ where d_S denotes the dimension of the state space and β_1 denotes the Hölder exponent that characterizes the smoothness of the Q-function. Similar result (i.e., optimal nonparametric convergence rate) can be obtained for the (conditional) probability ratio function (Wang et al. 2021). As discussed in Section 3.2.4, when T' and M are sufficiently large, (C3) essentially requires the estimated mean and covariance functions in the conditional Gaussian model to converge at a rate of $(NT)^{-\kappa_3}$. Under some regularity conditions, an optimal nonparametric convergence rate can also be achieved.

Condition (C4) is mild as the policy class Π is pre-specified. When a linear policy class is employed, we have $\kappa_4 = \#s$ where $\#s$ denotes the number of parameters used to index the policy class. When Π is set to some deep neural networks, the corresponding VC-dimension is also available in the literature (see e.g., Harvey, Liaw, and Mehrabian 2017).

The uniqueness of the optimal policy (C5) is commonly assumed in the literature (Ertefaie and Strawderman 2018; Luckett et al. 2020). The second part of (C5) is closely related to margin-type conditions commonly used to bound the excess misclassification error (Tsybakov et al. 2004; Audibert and Tsybakov 2007) and the regret of individualized treatment regimes in point treatment studies (Qian and Murphy 2011; Luedtke and Van Der Laan 2016; Shi, Lu, and Song 2020).

To better understand the margin condition in (C5), we first observe that $A^{\pi^{\text{opt}}}(a,s) \le 0$ for any a and s. To elaborate this, we note that π^{opt} maximizes $V^{\pi}(s)$ for any π. Consider the following history-dependent policy $\pi^{\text{opt}}(a)$ that assigns a at the initial decision point and follows π^{opt} in the subsequent steps. The value under such a policy is given by $Q^{\pi^{\text{opt}}}(a,s)$. It follows that $Q^{\pi^{\text{opt}}}(a,s) \le V^{\pi^{\text{opt}}}(s)$, or equivalently, $A^{\pi^{\text{opt}}}(a,s) \le 0$ for any a and s. The equality holds only when $a = \text{argmax}_{a'} Q^{\pi^{\text{opt}}}(a',s)$. The argmax is well-defined by the uniqueness of the optimal policy. For $a \ne \text{argmax}_{a'} Q^{\pi^{\text{opt}}}(a',s)$, the advantage function corresponds to the value difference between $\pi^{\text{opt}}(a)$ and π^{opt}. The smaller the difference, the harder it is to identify the optimal policy. To ensure π^{opt} can be consistently identified, it is thus reasonable to assume $\Pr(0 < |A^{\text{opt}}(a,S^*)| \le \epsilon)$ decays to zero with ϵ as well. (C5) explicitly characterizes such dependence. For example, when the action space is binary, let $\tau(s)$ denote the contrast function, that is, $\tau(s) = Q^{\pi^{\text{opt}}}(1,s) - Q^{\pi^{\text{opt}}}(0,s)$. It is immediate to see that $A^{\pi^{\text{opt}}}(0,s) = \min(-\tau(s),0)$ and $A^{\pi^{\text{opt}}}(1,s) = \min(\tau(s),0)$. Thus, the second part of (C5) essentially requires $\Pr(0 < |\tau(S^*)| \le \epsilon) \le \bar{c}\epsilon^{\alpha}$, which is automatically satisfied with $\alpha = 1$ when $\tau(S^*)$ has a bounded probability density function. More generally, it holds when $|\tau(S^*)|^{\alpha}$ has a bounded probability density function. For instance, suppose both the initial reference distribution ν and the Markov transition function have bounded density functions on $(0,+\infty)$. Then, the distribution of S^*, that is, the mixture distribution of $\{S_t\}_{t\ge 0}$ with weights $\{(1-\gamma)\gamma^t\}_{t\ge 0}$ has a bounded probability density function as well. Suppose the state is one-dimensional and $\tau(S^*) = (S^*)^{1/\alpha}$. Then it is immediate to see that $|\tau(S^*)|^{\alpha}$ has a bounded probability density function. Finally, when $|\tau(S^*)|$ is uniformly bounded away from zero, then (C5) holds with $\alpha = +\infty$. We will see in Theorem 2 that the convergence rate of π_{new} depends crucially on the margin parameter α.

Assumption (C6) characterizes the dependence of the data observations over time. It essentially requires the β-mixing coefficient (see e.g., Bradley 2005, for a detailed definition) at lag q, which measures the time dependence between the set of variables $\{(S_j,A_j,R_j)\}_{j\le t}$ and $\{(S_j,A_j,R_j)\}_{j\ge t+q}$, to decay to zero at an exponential rate with respect to q. This assumption automatically holds when $\{(S_t,A_t,R_t)\}_{t\ge 0}$ forms a geometrically ergodic Markov chain. Geometric ergodicity is less restrictive than those imposed in the existing reinforcement learning literature that requires observations to be independent (see e.g., Degris, White, and Sutton 2012; Farahmand et al. 2016) or to follow a uniform-ergodic Markov chain (see e.g., Bhandari, Russo, and Singal 2018).

Theorem 2 (Value Enhancement Property). Suppose (C1)–(C6) hold. If the constants $\kappa_1,\kappa_2,\kappa_3$ satisfy $\kappa_{i_1} + \kappa_{i_2} > 1/(2+2\alpha)$ for any disjoint $i_1, i_2 \in \{1,2,3\}$, and that $\mathcal{V}(\pi^{\text{opt}}) - \mathcal{V}(\pi_{\text{old}}^{(\ell)}) = O_p\{(NT)^{-\kappa_0}\}$ for any ℓ, we have $\mathcal{V}(\pi^{\text{opt}}) - \mathcal{V}(\pi_{\text{new}}) = E_1 + E_2$ where $E_1 = O_p\{(NT)^{-\frac{\kappa_0(2\alpha+1)}{\alpha+1}}\}$, $E_2 = o_p\{(NT)^{-1/2}\}$.

Theorem 2 states that the value difference $\mathcal{V}(\pi^{\text{opt}}) - \mathcal{V}(\pi_{\text{new}})$ can be decomposed into two terms. Here, the first term E_1 describes how the input policy π_{old} takes effect. It is due to the presence of the higher-order remainder term $\eta_2(\pi,\pi_{\text{old}})$ resulting from the first order approximation of the value difference $\mathcal{V}(\pi) - \mathcal{V}(\pi_{\text{old}})$. The second term E_2 is due to the estimation error of the $\eta_1(\pi,\pi_{\text{old}})$. In the typical multiply robust setting, it often requires that $\kappa_{i_1} + \kappa_{i_2} > 1/2$ so that the bias of estimating $\eta_1(\pi,\pi_{\text{old}})$ is $o_p\{(NT)^{-1/2}\}$. In Theorem 2, since we require slower rates for nuisance parameters, the proposed value difference estimator for $\eta_1(\pi,\pi_{\text{old}})$ may converge slower than the 1/2-root. However, the value enhancement property can still be established under such a slower rate requirement. This is due to that Theorem 2 is concerned with the convergence rate of the estimated optimal policy π_{new} in terms of the value instead of the rate of the proposed value difference estimator (denoted by $\widehat{\eta}_1(\pi_{\text{new}})$). In particular, the convergence rate of π_{new} in terms of the value is primarily determined by the difference between $\widehat{\eta}_1(\pi_{\text{new}})$ and $\widehat{\eta}_1(\pi^{\text{opt}})$ (see Page 12 of the supplementary material), which converges at a faster rate than $\widehat{\eta}_1(\pi_{\text{new}})$ itself. This is because π_{new} is consistent to the optimal

policy implied by the condition that $\mathcal{V}(\pi^{\text{opt}}) - \mathcal{V}(\pi_{\text{old}}^{(\ell)}) = O_p\{(NT)^{-\kappa_0}\}$.

When $\kappa_0 \leq 1/2$, it can be seen that the value under the output policy converges at a faster rate than the input policy, leading to the desired "value enhancement property". One can repeat the one-step update multiple times to guarantee that the value of the estimated optimal policy converges at a rate of $o_p\{(NT)^{-1/2}\}$. When the initial policy already converges faster than the parametric rate (e.g., $\kappa_0 > 1/2$), then our proposal is not guaranteed to yield a better policy in theory. However, as shown in our empirical studies (see Section 5), the values of the proposed policies are often larger than those computed via state-of-the-art RL algorithms. This suggests that although these initial policies are consistent, they might converge at a suboptimal rate and have room for improvement.

4.2. Efficiency of the Value Difference Estimator

In this section, we show that conditional on $\pi_{\text{old}}^{(\ell)}$, the proposed estimator for $\eta_1(\pi, \pi_{\text{old}}^{(\ell)})$, that is,

$$\widehat{\eta}_1(\pi, \pi_{\text{old}}^{(\ell)}) = \frac{1}{T|\mathcal{I}_\ell|} \times \left\{ \sum_{i \in \mathcal{I}_\ell} \sum_{t=0}^{T-1} \psi(O_{i,t}; \pi, \pi_{\text{old}}^{(\ell)}, \widehat{V}^{(\ell)}, \widehat{A}^{(\ell)}, \widehat{\omega}^{(\ell)}, \widehat{d}^{(\ell)}) \right\},$$

is nearly unbiased to $\eta_1(\pi, \pi_{\text{old}}^{(\ell)})$ and its asymptotic variance matches this efficiency bound. The notion of efficiency bound can be found in Section A.4 of supplementary material. Consequently, $\widehat{\eta}_1$ is efficient.

Let $\bar{p}$ denote the conditional distribution of (R_t, S_{t+1}) given (A_t, S_t). For any given π and π_{old}, we note that $\eta_1(\pi, \pi_{\text{old}})$ is completely determined by the transition function $\bar{p}$. Let $\{\bar{p}_{\theta_1} : \theta_1 \in \Theta_1\}$ be a regular parametric submodel for $\bar{p}$. This requires $\bar{p}_{\theta_1}$ to be a transition matrix for any θ_1 and $\bar{p} = \bar{p}_{\theta_1^*}$ for some $\theta_1^* \in \Theta_1$. Similarly, let $\{\bar{b}_{\theta_2} : \theta_2 \in \Theta_2\}$ and $\{\bar{\nu}_{\theta_3} : \theta_3 \in \Theta_3\}$ be regular parametric submodels for the behavior policy and the initial state distribution, respectively. Let $\theta = (\theta_1, \theta_2, \theta_3)$ and $\theta^* = (\theta_1^*, \theta_2^*, \theta_3^*)$ where θ_2^* and θ_3^* correspond to the true parameters in Θ_2 and Θ_3. Under a given submodel indexed by θ, the log-likelihood function of a single data trajectory can be written as

$$\ell_T(\{O_t\}_t; \theta) = \log \left[\bar{\nu}_{\theta_3}(S_0) \prod_{t=0}^{T} \{\bar{p}_{\theta_1}(R_t, S_{t+1}|A_t, S_t) \bar{b}_{\theta_2}(A_t|S_t)\} \right].$$

Note that η_1 can be defined as a function of θ as well. We define the efficiency bound as

$$\text{EB}(|\mathcal{I}_\ell|, T) = |\mathcal{I}_\ell| \sup \nabla_\theta \eta_1(\theta^*) \times \left\{ \mathbb{E} \nabla_\theta \ell_T(\{O_t\}_t; \theta^*) \nabla_\theta^\top \ell_T(\{O_t\}_t; \theta^*) \right\}^{-1} \times \nabla_\theta^\top \eta_1(\theta^*),$$

where the supremum is taken over all regular parametric submodels, and $\nabla_\theta g(\theta')$ denotes the derivative of a function g with respect to θ, evaluated at $\theta = \theta'$. As discussed before, η_1 depends on θ only through θ_1.

Theorem 3. Suppose the conditions in Theorem 2 holds with $\kappa_{i_1} + \kappa_{i_2} > 1/2$ for any disjoint $i_1, i_2 \in \{1, 2, 3\}$. Then conditional on $\pi_{\text{old}}^{(\ell)}$, we have for any π that

$$\frac{\widehat{\eta}_1(\pi, \pi_{\text{old}}^{(\ell)}) - \eta_1(\pi, \pi_{\text{old}}^{(\ell)})}{\sqrt{\text{EB}(|\mathcal{I}_\ell|, T)}} \xrightarrow{d} N(0, 1).$$

Theorem 3 implies that $\widehat{\eta}_1$ is asymptotically unbiased with asymptotic variance $\text{EB}(|\mathcal{I}_\ell|, T)$. This demonstrates the efficiency of the proposed estimator.

5. Numerical Examples

In this section, we use one toy example and real data related studies to demonstrate the superior performance of our method. Specifically, in Section 5.1, we use a toy example to demonstrate the multiple robustness of our estimator and the value enhancement property. We then demonstrate the performance of the proposed method on OhioT1DM related datasets in Section 5.2. In Appendix D, we conduct another simulation study to illustrate the finite-sample performance of our algorithm compared with several existing methods.

5.1. A Toy Example

We design a toy example to illustrate the multiple robustness of our estimator and the desired value enhancement property. Consider a binary state space $\mathcal{S} = \{0, 1\}$, where S_0 takes value 0 with probability 0.4 and otherwise. The action space $\mathcal{A}$ takes values in $\{0, 1\}$. The reward function is defined as $r(a, s) = \mathbb{I}(s = a)$, and then the reward is generated according to $R_t = r(A_t, S_t) + e_t$ where $\{e_t\}_{0 \leq t < T}$ is a sequence of iid $N(0, 2)$ random errors. The transition matrix of $p(S'|A, S)$ and behavior policy can be found in Section D of the supplementary material.

In this tabular case, the oracle values of Q^π, ω^π, and p can be simulated using Monte Carlo methods. To demonstrate the triply robustness property, we will add some random errors on Q^π, ω^π, or p to make them biased. Then we compute η_1 with these nuisance functions and obtain the resulting estimated optimal policy via our proposed algorithm. Specifically, we consider the following five combinations of nuisance function estimators: (i) "origin": all the nuisance functions are set to their oracle values. (ii) "mod1": ω^π is set to a biased value, while other nuisances are oracle; (iii) "mod2": Q^π is set to a biased value, while other nuisances are oracle. (iv) "mod3": The transition p is set to a biased value, while other nuisances are oracle. (vi) "mod4": all nuisance functions are set to bias values. Details of these scenarios can be found in Section D of the supplementary material.

To summarize, Scenario (i) corresponds to the oracle setting where all the nuisance functions are correctly specified. In Scenarios (ii)–(iv), one of the nuisance functions is misspecified. In the last scenario, all the nuisance functions are misspecified. We also vary the initial policy to investigate the value enhancement property. In particular, we represent the initial policy π_{old} using a 2×2 matrix and consider the following parameterization,

$$\pi_{\text{old}} = \begin{pmatrix} A = 0 & A = 1 \\ S = 0 & \kappa & 1 - \kappa \\ S = 1 & 1 - \kappa & \kappa \end{pmatrix},$$

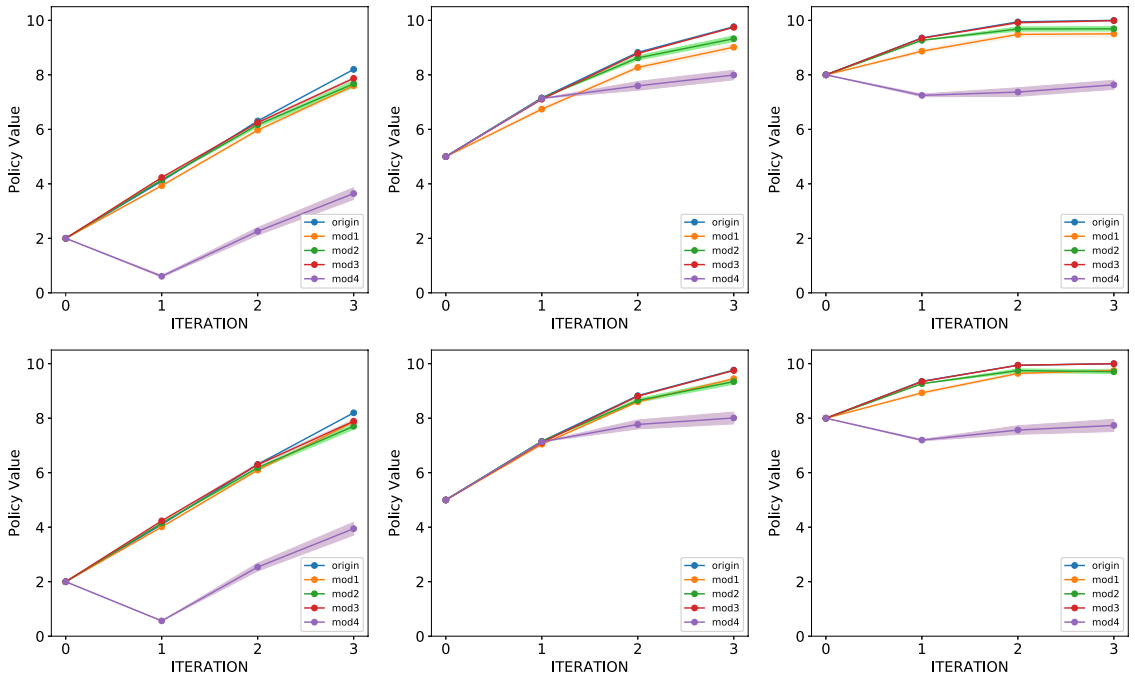

Figure 1. Values of estimated policies in a toy example. First row represents results using (T, N) pair as $(30, 30)$ while the second row using $(50, 50)$. The three columns represents initial policy factor κ taking values 0.8, 0.5, 0.2, respectively. The horizontal axis represents the number of iterations used in our value enhancement procedure. When iteration equals zero, we plot the evaluation value for the initial policy. The optimal value is 10 and δ is fixed to 0.1. The confidence band is computed based on 100 replications.

for some $\kappa \in [0, 1]$. According to our data generating mechanism, $\kappa = 0$ corresponds to the optimal policy. The closer κ is to 1, the worse the initial policy is. We consider three choices of κ, corresponding to 0.2, 0.5, and 0.8. This yields three different initial policies. We further consider two choices of sample size, $N = 30, T = 30$ and $N = 50, T = 50$. This yields a total of $5 \times 3 \times 2 = 30$ settings. γ is set to 0.9. It can be shown that the optimal value $\mathcal{V}(\pi^{\text{opt}})$ equals $(1 - \gamma)^{-1} = 10$. Finally, we consider three choices of δ (see (20)), corresponding to 0.05, 0.1 and 0.2, respectively.

Results are reported in Figures 1, S1, and S2 (see Appendix D in the supplementary article). All values of estimated policies are computed via Monte Carlo simulations. It can be observed that in Scenarios (i)–(iv), our proposed algorithm using models in (i)–(iv) substantially improves the performance of the initial policy, demonstrating the desired value enhancement property. In particular, when either one of nuisance functions models is misspecified, the proposed method remains valid. This empirically verifies the triply-robustness property.

In addition, when all models are misspecified, the proposed method is not guaranteed to improve the value. Specifically, in the first two columns, the proposed method under "mod4" improves the initial policy after a few iterations. In the last column, however, values of the estimated policies are smaller than the initial one. We suspect that this is because under model misspecification, our procedure may converge to a suboptimal policy whose value is bounded between the value of the initial policy in the second column and that in the last column. Consequently, when the existing policy given by other methods is already close to the optimal, using inconsistent estimators of nuisance functions could possibly degrade the performance. Finally, under settings where the initial policy is very different from the optimal one (i.e., the first columns of Figures 1, S1, and S2 in Appendix D), it requires more iterations and a larger δ for our method to achieve a larger value. In contrast, when the initial policy is close to the optimal one, fewer iterations are needed and a smaller δ would be preferred. For instance, it can be seen from the third column of Figure S2 in Appendix D that when $\delta = 0.2$, the values of the estimated policies using models in (ii) and (iii) decrease at the third iteration.

Finally, to further demonstrate the advantage of the proposed method, we use lookup tables (e.g., linear models with table lookup features) instead of deep learning models to parameterize all nuisance functions (including the Q-function, the probability ratio and the transition kernel), and apply the proposed method to this toy example. Results are reported in Figure S3 of the supplementary material. It can be seen that the proposed method is still able to improve the performance of initial policies.

5.2. Application to the OhioT1DM Related Datasets

There is an increasing interest in applying RL algorithms to mobile health(mHealth) applications. In this section, we conducted two analyses based on the OhioT1DM dataset. In the first analysis, we generate synthetic data to mimic this real dataset and apply our method to the synthetic dataset. In particular, we use the simulation environment designed in Section 5.2.2 of Shi et al. (2020a) for the data generation. In the second analysis, we apply our method to the real dataset.

The OhioT1DM dataset contains continuous measurements for six patents with type 1 diabetes over eight weeks. The state $\tilde{S}_t$ consists of three states, corresponding to the average blood glucose levels, the carbohydrate estimate for the meal

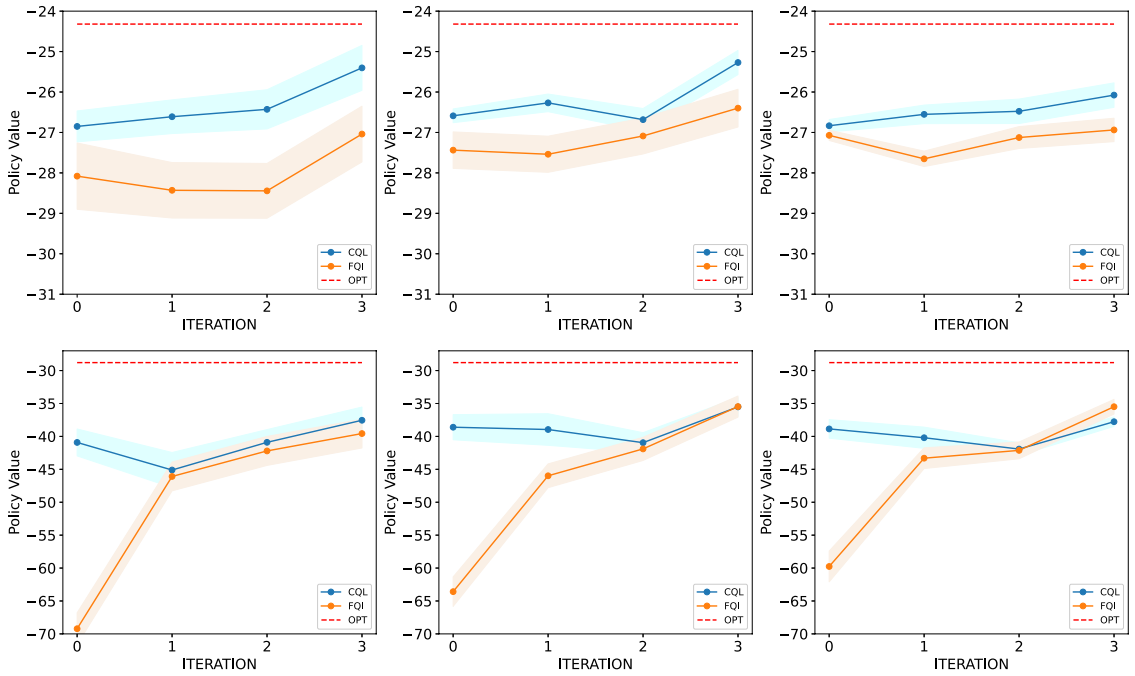

Figure 2. Values of various policies in the real data based simulation study. The initial policies are computed by CQL and FQI. The first row represents results using $\gamma = 0.9$ while the second row using $\gamma = 0.95$. The optimal values are equal to -24.32 and -28.79, respectively (shown by the read dash line). Three columns represents using (T, N) pair as (25, 100), (50, 50), (100, 25), respectively. The confidence band is computed based on 100 replications.

and the exercise intensity, respectively. The action A is the amount of insulin doses. We discretize the action space and consider five actions, that is, $\mathcal{A} = \{0, 1, 2, 3, 4\}$ from no to high doses of insulin. The Markov test developed by Shi et al. (2020a) suggests that the data are likely to satisfy a 4th order Markov property, so we reconstruct the state variable $S_t = (\tilde{S}_{t-3}, A_{t-3}, \tilde{S}_{t-2}, A_{t-2}, \tilde{S}_{t-1}, A_{t-1}, \tilde{S}_t)$ by concatenating past measurements to meet the Markov assumption. This yields to a 15-dimensional state vector. The reward R_t is defined as the Index of Glycemic Control that is a deterministic function of the average blood glucose levels during the time interval $[t, t + 1)$.

We first apply the proposed method to the synthetic datasets. We use FQI and CQL to compute the initial policy. We did not implement V-learning (VL) here since it requires large computational costs when the dataset is large. In Figure 2, it can be seen that we are able to achieve near-optimal policies after iterating the proposed algorithm three times. The estimated optimal policy achieves larger values than the initial policies in all cases. The improvement is substantial when π_{old} is not very close to the optimal policy.

We next apply our method to the real dataset. In order to evaluate the estimated optimal policy, we split the data into training and test datasets. After obtaining estimated optimal policies on the training data, we apply FQE on the test data to compute the policy values of all these estimated policies. Figure 3 reports these values. It implies that the proposed algorithm will yield a policy with larger value after 2–3 iterations. Lastly, we apply the estimated optimal policy based on the proposed algorithm to the whole dataset, with the initial policy computed by CQL. The overall proportion of recommending each action $(A = 0, 1, \ldots, 4)$ by our estimated policy is 15.2%, 0.5%, 2%, 6%, and 76%, respectively. The results imply that our estimated policy recommends the largest dose in most scenarios, with a certain proportion of recommending not receiving any insulin doses.

6. Discussion

In this article, we propose a value enhancement policy optimization method for offline RL problems. One of the key ingredients of the proposed methodology lies in developing a triply robust estimator for the first-order linear term η_1 which measures the difference between any two policies. There is a rich line of research on multiply robust estimators in causal inference. For instance, Tchetgen Tchetgen and Shpitser (2012) proposed triply robust estimators of the marginal natural indirect and direct effects in causal mediation analysis. Wang and Tchetgen Tchetgen (2018) and Shi et al. (2020c) developed triply robust estimators for the average treatment effect using instrumental variables and double negative control variables, respectively. Jiang, Yang, and Ding (2020) proposed triply robust estimators for the causal effects within principal strata. Our proposed estimator shares similar statistical properties to these estimators in that its consistency only requires two out of three nuisance functions to be correctly specified. In addition, it is efficient when all functions are correctly specified and satisfy certain convergence rates.

Based on the triply robust estimator, we propose to search an optimal policy that maximizes the value difference subject to a trust region constraint, and iterate this procedure for value enhancement. We discuss the choice of the number of iterations and the computation time of our proposal in Section E of the supplementary material. Finally, the proposed method can be used as a stand-alone policy iteration algorithm that starts with a completely random initial policy and iteratively updates this policy to improve its performance. However, the resulting algorithm can be computationally intensive in practice, since it

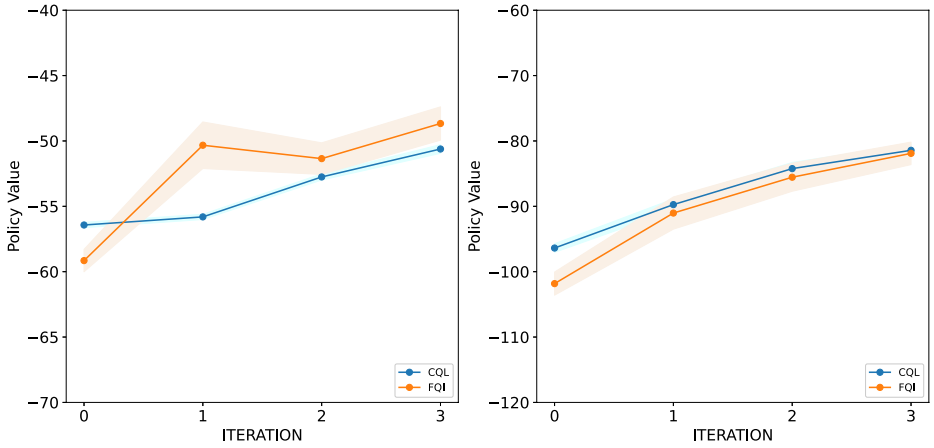

Figure 3. Values of various policy computed based on the real dataset. The initial policies are computed by CQL and FQI. Left figure corresponds to the result of $\gamma = 0.9$ while the right one considers $\gamma = 0.95$. The policy values are computed by cross-validation procedure and the confidence band is computed based on 100 replications.

might require a large number of iterations to achieve a near-optimal policy.

Supplementary Materials

The supplementary materials provide proofs for all technical lemmas, theorems, corollaries, details on the proposed algorithm, and additional numerical studies.

Acknowledgment

The authors thank the Editor, Associate Editor, and anonymous reviewers for their suggestions and helpful feedback which improved the paper significantly.

Disclosure Statement

No potential conflict of interest was reported by the author(s).

Funding

Dr. Fan Zhou's work is supported by National Natural Science Foundation of China (12001356), "Chenguang Program" supported by Shanghai Education Development Foundation and Shanghai Municipal Education Commission, Open Research Projects of Zhejiang Lab (No. 2022RC0AB06), Shanghai Research Center for Data Science and Decision Technology, Innovative Research Team of Shanghai University of Finance and Economics.

References

Abbeel, P., and Ng, A. Y. (2004), "Apprenticeship Learning via Inverse Reinforcement Learning," in *Proceedings of the Twenty-First International Conference on Machine Learning*, p. 1. [2011]

Audibert, J.-Y., and Tsybakov, A. B. (2007), "Fast Learning Rates for Plug-in Classifiers," *The Annals of Statistics*, 35, 608–633. [2020]

Bertsekas, D. P., and Tsitsiklis, J. N. (1996), *Neuro-Dynamic Programming* (Vol. 5), Belmont, MA: Athena Scientific. [2017]

Bhandari, J., Russo, D., and Singal, R. (2018), "A Finite Time Analysis of Temporal Difference Learning with Linear Function Approximation," arXiv preprint arXiv:1806.02450. [2020]

Bishop, C. (1994), "Mixture Density Networks," *Technical Report*, pp. 1–26. [2018]

Bradley, R. C. (2005), "Basic Properties of Strong Mixing Conditions. A Survey and Some Open Questions," *Probability Surveys*, 2, 107–144. [2020]

Chakraborty, B., and Moodie, E. (2013), *Statistical Methods for Dynamic Treatment Regimes*, New York: Springer. [2011]

Chen, X., and Qi, Z. (2022), "On Well-posedness and Minimax Optimal Rates of Nonparametric q-function Estimation in Off-policy Evaluation," in *International Conference on Machine Learning*, PMLR, pp. 3558–3582. [2020]

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., and Robins, J. (2018), "Double/Debiased Machine Learning for Treatment and Structural Parameters," *The Econometrics Journal*, 21, C1–C68. [2015,2016]

Chernozhukov, V., Chetverikov, D., and Kato, K. (2014), "Gaussian Approximation of Suprema of Empirical Processes," *The Annals of Statistics*, 42, 1564–1597. [2019]

Degris, T., White, M., and Sutton, R. S. (2012), "Off-Policy Actor-Critic," in *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pp. 179–186. [2020]

Ernst, D., Geurts, P., Wehenkel, L., and Littman, L. (2005), "Tree-based Batch Mode Reinforcement Learning," *Journal of Machine Learning Research*, 6, 503–556. [2012]

Ertefaie, A., and Strawderman, R. L. (2018), "Constructing Dynamic Treatment Regimes over Indefinite Time Horizons," *Biometrika*, 105, 963–977. [2012,2020]

Fan, J., Wang, Z., Xie, Y., and Yang, Z. (2020a), "A Theoretical Analysis of Deep q-Learning," in *Learning for Dynamics and Control*, PMLR, pp. 486–489. [2012]

——— (2020b), "A Theoretical Analysis of Deep q-learning," in *Learning for Dynamics and Control*, PMLR, pp. 486–489. [2020]

Farahmand, A.-m., Ghavamzadeh, M., Szepesvári, C., and Mannor, S. (2016), "Regularized Policy Iteration with Nonparametric Function Spaces," *The Journal of Machine Learning Research*, 17, 4809–4874. [2020]

Feng, Y., Ren, T., Tang, Z., and Liu, Q. (2020), "Accountable Off-Policy Evaluation with Kernel Bellman Statistics," in *International Conference on Machine Learning*, PMLR, pp. 3102–3111. [2020]

Harvey, N., Liaw, C., and Mehrabian, A. (2017), "Nearly-Tight vc-dimension Bounds for Piecewise Linear Neural Networks," in *Conference on Learning Theory*, PMLR, pp. 1064–1068. [2020]

Hu, X., Qian, M., Cheng, B., and Cheung, Y. K. (2021), "Personalized Policy Learning using Longitudinal Mobile Health Data," *Journal of the American Statistical Association*, 116, 410–420. [2012]

Hubbs, C. D., Perez, H. D., Sarwar, O., Sahinidis, N. V., Grossmann, I. E., and Wassick, J. M. (2020), "Or-gym: A Reinforcement Learning Library for Operations Research Problem," arXiv preprint arXiv:2008.06319 . [2011]

Hunter, D. R., and Lange, K. (2004), "A Tutorial on MM Algorithms," *The American Statistician*, 8, 30–37. [2014]

Jiang, Z., Yang, S., and Ding, P. (2020), "Multiply Robust Estimation of Causal Effects Under Principal Ignorability," arXiv preprint arXiv:2012.01615. [2023]

Kakade, S., and Langford, J. (2002), "Approximately Optimal Approximate Reinforcement Learning," in *ICML* (Vol. 2), pp. 267–274. [2012,2014]

Kallus, N., and Uehara, M. (2019), "Efficiently Breaking the Curse of Horizon: Double Reinforcement Learning in Infinite-Horizon Processes," arXiv preprint arXiv:1909.05850. [2014,2015,2020]

Kallus, N., and Uehara, M. (2020), Statistically Efficient Off-Policy Policy Gradients," in *International Conference on Machine Learning*, PMLR, pp. 5089–5100. [2012]

Komorowski, M., Celi, L. A., Badawi, O., Gordon, A. C., and Faisal, A. A. (2018), "The Artificial Intelligence Clinician Learns Optimal Treatment Strategies for Sepsis in Intensive Care," *Nature Medicine*, 24, 1716–1720. [2011]

Kosorok, M. R., and Laber, E. B. (2019), "Precision Medicine," *Annual Review of Statistics and its Application*, 6, 263–286. [2012]

Kumar, A., Zhou, A., Tucker, G., and Levine, S. (2020), "Conservative q-learning for Offline Reinforcement Learning," arXiv preprint arXiv:2006.04779. [2015,2017]

Laber, E. B., Lizotte, D. J., Qian, M., Pelham, W. E., and Murphy, S. A. (2014), "Dynamic Treatment Regimes: Technical Challenges and Applications," *Electronic Journal of Statistics*, 8, 1225–1272. [2012]

Le, H., Voloshin, C., and Yue, Y. (2019), "Batch Policy Learning Under Constraints," in *International Conference on Machine Learning*, pp. 3703–3712. [2017]

LeCun, Y., Bengio, Y., and Hinton, G. (2015), "Deep Learning," *Nature*, 521, 436–444. [2017]

Levine, S., Kumar, A., Tucker, G., and Fu, J. (2020), "Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems," arXiv preprint arXiv:2005.01643. [2019]

Li, Y. (2017), "Deep Reinforcement Learning: An Overview," arXiv preprint arXiv:1701.07274 . [2011]

Liao, P., Klasnja, P., and Murphy, S. (2019), "Off-Policy Estimation of Long-Term Average Outcomes with Applications to Mobile Health," arXiv preprint arXiv:1912.13088. [2012]

Liao, P., Qi, Z., and Murphy, S. (2020), "Batch Policy Learning in Average Reward Markov Decision Processes," arXiv preprint arXiv:2007.11771. [2012]

Liu, Q., Li, L., Tang, Z., and Zhou, D. (2018), "Breaking the Curse of Horizon: Infinite-Horizon Off-policy Estimation," in *Advances in Neural Information Processing Systems*, pp. 5356–5366. [2015,2017]

Luckett, D. J., Laber, E. B., Kahkoska, A. R., Maahs, D. M., Mayer-Davis, E., and Kosorok, M. R. (2020), "Estimating Dynamic Treatment Regimes in Mobile Health Using v-learning," *Journal of the American Statistical Association*, 115, 692–706. [2012,2017,2020]

Luedtke, A. R., and Van Der Laan, M. J. (2016), "Statistical Inference for the Mean Outcome Under a Possibly Non-unique Optimal Treatment Strategy," *Annals of Statistics*, 44, 713–742. [2020]

Marcolino, M. S., Oliveira, J. A. Q., D'Agostino, M., Ribeiro, A. L., Alkmim, M. B. M., and Novillo-Ortiz, D. (2018), "The Impact of mhealth Interventions: Systematic Review of Systematic Reviews," *JMIR mHealth and uHealth*, 6, e23. [2011]

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*, 518, 529–533. [2012]

Murphy, S. A. (2003), "Optimal Dynamic Treatment Regimes," *Journal of the Royal Statistical Society*, Series B, 65, 331–355. [2011]

Puterman, M. L. (1994), *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, Hoboken, NJ: Wiley. [2012,2013]

Qian, M., and Murphy, S. A. (2011), "Performance Guarantees for Individualized Treatment Rules," *Annals of Statistics*, 39, 1180–1210. [2011,2020]

Rust, J. (1987), "Optimal Replacement of GMC Bus Engines: An Empirical Model of Harold Zurcher," *Econometrica: Journal of the Econometric Society*, 55, 999–1033. [2011]

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015), "Trust Region Policy Optimization," *International Conference on Machine Learning*, pp. 1889–1897. [2012,2013,2014,2015]

Shi, C., Fan, A., Song, R., and Lu, W. (2018), "High-Dimensional a-learning for Optimal Dynamic Treatment Regimes," *Annals of Statistics*, 46, 925. [2011]

Shi, C., Lu, W., and Song, R. (2020), "Breaking the Curse of Nonregularity with Subagging: Inference of the Mean Outcome Under Optimal Treatment Regimes," *Journal of Machine Learning Research*, accepted. [2020]

Shi, C., Wan, R., Chernozhukov, V., and Song, R. (2021), "Deeply-Debiased Off-Policy Interval Estimation," in *Proceedings of the 38th International Conference on Machine Learning*, Vol. 139 of Proceedings of Machine Learning Research, PMLR, pp. 9580–9591. [2014,2017]

Shi, C., Wan, R., Song, R., Lu, W., and Leng, L. (2020a), "Does the Markov Decision Process Fit the Data: Testing for the Markov Property in Sequential Decision Making," in *International Conference on Machine Learning*, PMLR, pp. 8807–8817. [2013,2022,2023]

Shi, C., Zhang, S., Lu, W., and Song, R. (2020b), "Statistical Inference of the Value Function for Reinforcement Learning in Infinite Horizon Settings," arXiv preprint arXiv:2001.04515 . [2011,2020]

Shi, X., Miao, W., Nelson, J. C., and Tchetgen Tchetgen, E. J. (2020c), "Multiply Robust Causal Inference with Double-Negative Control Adjustment for Categorical Unmeasured Confounding," *Journal of the Royal Statistical Society*, Series B, 82, 521–540. [2023]

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M. et al. (2016), "Mastering the Game of Go with Deep Neural Networks and Tree Search," *Nature*, 529, 484–489. [2011]

Sutton, R. S., and Barto, A. G. (2018), *Reinforcement Learning: An Introduction*, Cabridge, MA: MIT Press. [2011,2012,2013,2017]

Tchetgen Tchetgen, E. J., and Shpitser, I. (2012), "Semiparametric Theory for Causal Mediation Analysis: Efficiency Bounds, Multiple Robustness, and Sensitivity Analysis," *Annals of Statistics*, 40, 1816. [2023]

Tsiatis, A. A., Davidian, M., Holloway, S. T., and Laber, E. B. (2019), *Dynamic Treatment Regimes: Statistical Methods for Precision Medicine*, Boca Raton, FL: CRC Press. [2012]

Tsybakov, A. B. (2004), "Optimal Aggregation of Classifiers in Statistical Learning," *The Annals of Statistics*, 32, 135–166. [2020]

Wang, J., Qi, Z., and Wong,R. K. (2021), "Projected State-Action Balancing Weights for Offline Reinforcement Learning," arXiv preprint arXiv:2109.04640 . [2020]

Wang, L., and Tchetgen Tchetgen, E. (2018), "Bounded, Efficient and Multiply Robust Estimation of Average Treatment Effects Using Instrumental Variables," *Journal of the Royal Statistical Society*, Series B, 80, 531–550. [2023]

Wang, L., Zhou, Y., Song, R., and Sherwood, B. (2018), "Quantile-Optimal Treatment Regimes," *Journal of the American Statistical Association*, 113, 1243–1254. [2011]

Watkins, C. J., and Dayan, P. (1992), "Q-learning," *Machine Learning*, 8, 279–292. [2012]

Wu, Y., Tucker, G., and Nachum, O. (2019), "Behavior Regularized Offline Reinforcement Learning," arXiv preprint arXiv:1911.11361 . [2019]

Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J. Y., Levine, S., Finn, C., and Ma, T. (2020), "Mopo: Model-based Offline Policy Optimization," *Advances in Neural Information Processing Systems*, 33, 14129–14142. [2018]

Zhao, Y.-Q., Zeng, D., Laber, E. B., and Kosorok, M. R. (2015), "New Statistical Learning Methods for Estimating Optimal Dynamic Treatment Regimes," *Journal of the American Statistical Association*, 110, 583–598. [2011]