HHS Public Access

Author manuscript

Ann Stat. Author manuscript; available in PMC 2023 April 04.

Published in final edited form as:

Ann Stat. 2022 December ; 50(6): 3364–3387. doi:10.1214/22-aos2231.

BATCH POLICY LEARNING IN AVERAGE REWARD MARKOV DECISION PROCESSES

Peng Liao[*,¶], **Zhengling Qi**[†], **Runzhe Wan**[‡], **Predrag Klasnja**[§], **Susan A. Murphy**[*]

[*]Harvard University

[†]George Washington University

[‡]Amazon

[§]University of Michigan

Abstract

We consider the batch (off-line) policy learning problem in the infinite horizon Markov Decision Process. Motivated by mobile health applications, we focus on learning a policy that maximizes the long-term average reward. We propose a doubly robust estimator for the average reward and show that it achieves semiparametric efficiency. Further we develop an optimization algorithm to compute the optimal policy in a parameterized stochastic policy class. The performance of the estimated policy is measured by the difference between the optimal average reward in the policy class and the average reward of the estimated policy and we establish a finite-sample regret guarantee. The performance of the method is illustrated by simulation studies and an analysis of a mobile health study promoting physical activity.

Keywords

Markov Decision Process; Average Reward; Policy Optimization; Doubly Robust Estimator

1. Introduction.

Mobile health (mHealth) is a rapidly growing field due to the recent advances in mobile and sensing technologies. The mHealth intervention provides a unique opportunity to promote the healthy behaviors (e.g., regular physical activity and adherence to medications) and has been successfully applied in many health fields (e.g., smoking cessation, physical activity, drug abuse and diabetes). Just-in-time adaptive interventions (JITAI, Nahum-Shani et al. (2016)) use a decision rule (i.e., a treatment policy or policy) that maps real-time information about the individual's context to a particular treatment. In this work we study the problem of how to use data consisting of multiple trajectories to estimate a policy that leads to good long-term performance.

We model the sequential decision making process by a time-homogeneous Markov Decision Process (MDP) (Puterman, 1994) over infinite time horizon. This framework is natural

[¶]The first two authors contributed equally.

for mobile health applications in which the number of decision times is often large. For example, in *HeartSteps*, a physical activity mHealth study, there are five decision times per day, resulting in thousands of decision times over a year. Tremendous progress has been made in finite horizon setting; see the recent review by Kosorok and Laber (2019) for references therein. However when the number of time points is very large, methods that are based on the idea of backward iteration (e.g., Q-learning) or importance sampling (Precup, 2000) may suffer a large variance in problems or even be unpractical (Voloshin et al., 2019; Laber et al., 2014).

We propose to estimate the policy that optimizes the long-term average outcomes (rewards) using data consisting of multiple trajectories of finite length. The majority of existing methods focuses on the alternative, the discounted sum of rewards (Sutton and Barto, 2018); see the recent works in statistics (Luckett et al., 2019; Ertefaie and Strawderman, 2018; Shi et al., 2020, 2021). The discounted formulation weighs immediate rewards more heavily than rewards further in the future, which is practical in some applications (e.g., finance). However, for mHealth applications, choosing an appropriate discount rate could be non-trivial. The rewards (i.e., the health outcomes) in the distant future are as important as the near-term ones, especially when considering maintenance of health behaviors as well as longer term treatment burden. This suggests using a large discount rate. However, it is well known that algorithms developed in the discounted setting can become increasingly unstable as the discount rate goes to one; see for example Naik et al. (2019). The long-term average reward framework provides a good approximation to the long-term performance of a desired treatment policy in mHealth. Indeed, it can be shown that under regularity conditions the finite average of the expected rewards converges sublinearly to the long-term average reward as time goes to infinity (Hernández-Lerma and Lasserre, 1999). Therefore, a policy that optimizes the average reward would approximately maximize the sum of the rewards over a sufficiently long time horizon.

In this work, we present a novel algorithm that estimates the optimal policy in a prespecified, parametric policy class. Various methods have been proposed to estimate the global optimal policy by estimating the optimal Q-function;see for example Ormoneit and Sen (2003); Lagoudakis and Parr (2003); Ernst et al. (2005); Munos and Szepesvári (2008); Antos, Szepesvári and Munos (2008a,b); Ertefaie and Strawderman (2018); Fujimoto, Meger and Precup (2019); Kumar et al. (2019); Agarwal, Schuurmans and Norouzi (2020). In practice, the optimal Q-function could be highly non-smooth and complex, thus requiring the use of a flexible function class. This usually results in a learned policy that is also complex. If interpretability is important, this is problematic. Furthermore, when the training data is limited, the flexible function class might overfit the data and thus the variance of the estimated value function and the corresponding policy could be high. Restricting to a pre-specified policy class was studied by Zhang et al. (2012, 2013); Zhou et al. (2017); Zhao et al. (2015, 2019); Athey and Wager (2017) in finite time horizon problems and by Luckett et al. (2019); Murphy et al. (2016); Liu et al. (2019) in infinite time horizon problems. The restriction to a simple policy class enhance the interpretability of the learned policy and reduces the variance of the learned policy, although this induces a bias when the optimal policy is not in the class (i.e., trading off the bias and variance).

To efficiently learn an optimal policy in a prespecified policy class, the main statistical challenge is to construct an estimator for the average reward of a policy that is both data-efficient and performs uniformly well when optimizing over the policy class. Our first contribution of this work is a novel doubly robust estimator (see Section 3); we show that this estimator achieves the semiparametric efficiency bound under certain conditions on the estimation error of nuisance functions (see Section 5). Doubly robust estimators have been developed in the finite time horizon problems (Robins, Rotnitzky and Zhao, 1994; Murphy et al., 2001; Dudík et al., 2014; Jiang and Li, 2016; Thomas and Brunskill, 2016) and recently in the discounted reward infinite horizon setting (Kallus and Uehara, 2019a; Tang et al., 2020). To the best of our knowledge, our doubly robust estimator for the long-term average reward is new. In the literature of the average MDP in the batch setting, only the non-doubly robust estimator proposed by Liao, Klasnja and Murphy (2019) for the long-term average reward can be shown to achieve the semi-parametric efficiency, although they did not explicitly derive it. Most of the previous works on the policy optimization/evaluation under this framework are focused the online setting or under parametric models (e.g., Mahadevan, 1996; Abounadi, Bertsekas and Borkar, 2001; Wan, Naik and Sutton, 2021). Theoretical studies on the average reward MDP in the batch setting are very limited, especially under non-parametric models.

To establish the semiparametric efficiency of the doubly robust estimator and the regret bound, we derive finite-sample error bounds for two nuisance function estimators, a relative value estimator and a ratio estimator. The obtained error bounds are shown to hold uniformly over the prespecified class of policies. Both the relative value and ratio estimators are both derived from the same principle (i.e., coupled estimation; see Section 4). In the case of the ratio estimator, we use an iterative procedure to obtain a near-optimal error bound for the ratio estimator. To the best of our knowledge, this is the first theoretical result characterizing the ratio estimation error, which might be of independent interest.

We learn the optimal policy by maximizing the estimated average reward over a policy class and derive a finite-sample upper bound of the regret. We show that the our proposed method achieves $O(p^{1/2}n^{-1/2} + pn^{-\tilde{\beta}})$ regret, where p is the number of parameters in the policy, n is the number of trajectories in the training data and $\tilde{\beta}$ is a constant that can be chosen arbitrarily close to $1/(1 + \alpha)$. Here $\alpha \in (0, 1)$ measures the complexity of nuisance function classes. The use of doubly robust estimation ensures the estimation error for the nuisance functions contributes only lower order terms to the regret. Unlike the setting in which the goal is to maximize the average reward defined over a finite horizon (Athey and Wager, 2017), when the goal is to maximize the average reward defined over an infinite horizon, the regret analysis requires uniform control of the estimation error of the policy-dependent nuisance functions over the policy class. We believe this is the first regret bound result for an estimator of an in-class optimal policy in the average reward MDP and using batch data. Recently, Sharma, Jafarnia-Jahromi and Jain (2020) proposed an approximate relative value learning algorithm for globally optimal policies under the average reward MDP with non-parametric function approximation. However, they require the sample size at least exponentially larger than the dimension of the state for the convergence of their algorithm, which seems sub-optimal compared with our result, e.g., Theorem 5.1.

The rest of the article is organized as follows. Section 2 formalizes the decision making problem and introduces the average reward MDP. Section 3 presents the proposed method of learning the in-class optimal policy, including the doubly robust estimator for average reward (Section 3.3). In Section 4, the coupled estimators of the policy-dependent nuisance functions are introduced. Section 5 provides a thorough theoretical analysis on the regret bound of our proposed method. In Section 6, we describe a practical optimization algorithm when Reproducing Kernel Hilbert Spaces (RKHSs) are used to model the nuisance functions. We further conduct several simulation studies to demonstrate the promising performance of our method in Section 7. All the technical proofs are postponed to the supplementary material.

2. Problem Setup.

Suppose we observe a training dataset, $\mathcal{D}_n = \{D_i\}_{i=1}^n$ that consists of n independent, identically distributed (i.i.d.) observations of D:

$$D = \{S_1, A_1, S_2, \ldots, S_T, A_T, S_{T+1}\}.$$

We use t to index the decision time. The length of the trajectory, T, is a fixed constant. $S_t \in \mathcal{S}$ is the state at time t and $A_t \in \mathcal{A}$ is the action (treatment) selected at time t. We assume the action space, $\mathcal{A}$, is finite. To eliminate unnecessary technical distractions, we assume that the state space, $\mathcal{S}$, is finite; this assumption imposes no practical limitations and can be extended to the general state space.

The states evolve according to a time-homogeneous Markov process, that is, for $t \geq 1$, $S_{t+1} \perp \{S_1, A_1, \ldots, S_{t-1}, A_{t-1}\} | \{S_t, A_t\}$, and the conditional distribution does not depend on t. Denote the conditional distribution by P, i.e., $\Pr(S_{t+1} = s' | S_t = s, A_t = a) = P(s' | s, a)$. The reward (i.e., outcome) is denoted by R_{t+1}, which is assumed to be a known function of (S_t, A_t, S_{t+1}), i.e., $R_{t+1} = \mathcal{R}(S_t, A_t, S_{t+1})$. We assume the reward is bounded, i.e., $|\mathcal{R}(s, a, s')| \leq R_{\max}$. We use $r(s,a)$ to denote the conditional expectation of reward given state and action, i.e., $r(s, a) = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$.

Let $H_t = \{S_1, A_1, \ldots, S_t\}$ be the history up to time $(t-1)$ and the current state, S_t. Denote the conditional distribution of A_t given H_t by $\pi_{b,t}(a|H_t) = \Pr(A_t = a|H_t)$. Let $\pi_b = \{\pi_{b,1}, \ldots, \pi_{b,T}\}$. This is often called behavior policy in the literature. Throughout this paper, the expectation, $\mathbb{E}$, without any subscript is assumed taken with respect to the distribution of the trajectory, D, with the actions selected by the behavior policy π_b.

Consider a time-stationary, Markovian policy, π, that takes the state as input and outputs a probability distribution on the action space, $\mathcal{A}$, that is, $\pi(a|s)$ is the probability of selecting action, a, at state, s. The average reward of the policy, π, is defined as

$$\eta^{\pi}(s) := \lim_{t^* \to \infty} \mathbb{E}_{\pi}\left(\frac{1}{t^*}\sum_{t=1}^{t^*} R_{t+1} \mid S_1 = s\right), \tag{2.1}$$

where the expectation, $\mathbb{E}_\pi$, is with respect to the distribution of the trajectory in which the states evolve according to P and the actions are chosen by $\boldsymbol{\pi}$. In the time-homogeneous MDP with finite state and bounded reward, the limit in (2.1) always exists (Puterman, 1994). The policy, $\boldsymbol{\pi}$, induces a Markov chain of states with the transition as $P^\pi(s'|s) = \sum_a \pi(a|s)P(s'|s,a)$. When the induced Markov chain, $P^{\boldsymbol{\pi}}$, is irreducible, it can be shown (e.g., in Puterman (1994)) that the stationary distribution of $P^{\boldsymbol{\pi}}$ exists and is unique (denoted by $d^{\boldsymbol{\pi}}$) and the average reward, $\eta^{\boldsymbol{\pi}}(s)$ (2.1) is independent of initial state (denoted by $\eta^{\boldsymbol{\pi}}$) and equal to

$$\eta^\pi(s) = \eta^\pi = \sum_{s,a} r(s,a)\pi(a|s)d^\pi(s)\,. \tag{2.2}$$

Throughout this paper we consider only the time-stationary, Markovian policies. In fact, it can be shown that the maximal average reward among all possible history dependent policies can be in fact achieved by some time-stationary, Markovian policy (Theorem 8.1.2 in Puterman (1994)). Consider a pre-specified class of such policies, Π, that is parameterized by $\theta \in \Theta \subset \mathbb{R}^p$. Throughout we assume that the induced Markov chain is always irreducible for any policy in the class, which is summarized below.

Assumption 1.

The induced Markov chain, $P^{\boldsymbol{\pi}}$, is irreducible for $\boldsymbol{\pi} \in \Pi$.

The goal of this paper is to develop a method that can efficiently use the training data, $\mathcal{D}_n$, to learn a policy that maximizes the average reward over Π. We propose to construct $\hat{\eta}_n^\pi$, an efficient estimator for the average reward, $\eta^{\boldsymbol{\pi}}$, for each policy $\boldsymbol{\pi} \in \Pi$ and learn an optimal policy by solving

$$\hat{\pi}_n \in \operatorname{argmax}_{\pi \in \Pi} \hat{\eta}_n^\pi\,. \tag{2.3}$$

The performance of $\hat{\pi}_n$ is measured by its regret, defined as

$$\operatorname{Regret}(\hat{\pi}_n) = \sup_{\pi \in \Pi} \eta^\pi - \eta^{\hat{\pi}_n}\,. \tag{2.4}$$

Note that although the average reward of the learned policy, $\hat{\pi}_n$, is defined over an infinite horizon, the goal here is to characterize the regret based on using a finite number of trajectories, n, hence the finite sample regret bound is in terms of n. Indeed while the average reward, $\eta^{\boldsymbol{\pi}}$ is defined as $t^* \to \infty$ (2.1), the Markovian and stationary assumptions allow us to estimate $\eta^{\boldsymbol{\pi}}$ using fixed length trajectories.

3. Doubly Robust Estimator for Average Reward.

In this section we present a doubly robust estimator for the average reward for a given policy. The estimator is derived from the efficient influence function (EIF). Below we first introduce two functions that occur in the EIF of the average reward. Throughout this section we fix a time-stationary Markovian policy, $\boldsymbol{\pi}$, and focus on the setting where the induced Markov chain, $P^{\boldsymbol{\pi}}$, is irreducible (Assumption 1).

3.1. Relative value and ratio functions.

First, we define the relative value function by

$$Q^{\pi}(s,a) := \lim_{t^* \to \infty} \frac{1}{t^*} \sum_{t=1}^{t^*} \mathbb{E}_{\pi}\left[\sum_{k=1}^{t} (R_{k+1} - \eta^{\pi}) \middle| S_1 = s, A_1 = a \right]. \tag{3.1}$$

The above limit is well-defined (Puterman (1994), p. 338). If we further assume the induced Markov chain is aperiodic, then the Cesàro limit in (3.1) can be replaced by $Q^{\pi}(s,a) = \mathbb{E}_{\pi}\{\sum_{t=1}^{\infty}(R_{t+1} - \eta^{\pi}) | S_1 = s, A_1 = a\}$. Q^{π} is often called relative value function in that $Q^{\pi}(s, a)$ represents the expected total difference between the reward and the average reward under the policy, π, when starting at state, s, and action, a.

The relative value function, Q^{π}, and the average reward, η^{π}, are closely related via the Bellman equation:

$$\mathbb{E}_{\pi}\left[R_{t+1} + Q(S_{t+1}, A_{t+1}) | S_t = s, A_t = a\right] - Q(s,a) - \eta = 0. \tag{3.2}$$

Note that in the above expectation $A_{t+1} \sim \pi(\cdot|S_{t+1})$. It is known that under the irreducibility assumption, the set of solutions of (3.2) is given by $\{(\eta^{\pi}, Q): Q = Q^{\pi} + c\mathbf{1}, c \in \mathbb{R}\}$ where $\mathbf{1}(s, a) = 1$ for all (s, a); see Puterman (1994), p. 343 for details. As we will see in Section 4.2, the Bellman equation provides the foundation of estimating the relative value function.

We now introduce the ratio function. For $t = 1, \ldots, T$, let $d_t(s, a)$ be the probability mass of state-action pair at time t in the trajectory D generated by the behavior policy. Denote by $d_D(s,a) := (1/T)\sum_{t=1}^{T} d_t(s,a)$ the average probability mass across the T decision times in D. Similarly, define $d_t(s)$ as the marginal distribution of S_t and $d_D(s) := (1/T)\sum_{t=1}^{T} d_t(s)$ as the average distribution of states in the trajectory D. Recall that T is the fixed length of the trajectory, D; d_D describes the distribution of this finite length trajectory. Further recall that under Assumption 1, the stationary distribution of P^{π} exists and is denoted by $d^{\pi}(s)$. We assume the following conditions on the data-generating process.

Assumption 2.—The data-generating process satisfies:

- **(2–1)** There exists some $p_{\min} > 0$, such that $\pi_{b,t}(a|H_t) \geq p_{\min}$ for all $a \in \mathcal{A}$ and $1 \leq t \leq T$ almost surely.
- **(2–2)** The average distribution $d_D(s) > 0$ for all $s \in \mathcal{S}$.

Under Assumption 2, it is easy to see that $d_D(s, a) \geq p_{\min} \cdot (\min_s d_D(s)) > 0$ for all state-action pair, (s, a). It essentially states that the data generating process ensures that every state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$ has a positive probability of being visited, which is a standard assumption in the literature. See e.g., Theorem 7 of Kallus and Uehara (2019b) and (A2) of Shi et al. (2020). In particular, Assumption (2–1) is often satisfied in randomized trials. See our mobile health application in Section 8. In addition, the batch data of our mobile health application consist of 37 trajectories with 210 decision points on each trajectory. In this application, as long as every state has a positive probability of being visited in at least one of 210 decision points, Assumption (2–2) is also satisfied. Assumption

(2–2) is imposed on the data generating process. We essentially require that across an infinite number of draws from this data generating process/trajectory, every state $s \in \mathcal{S}$ will be observed. Note that Assumption 2 does not require that the form of the behavior policy is known. Now we can define the ratio function:

$$\omega^{\pi}(s,a) = \frac{d^{\pi}(s)\pi(a\,|\,s)}{d_D(s,a)} \tag{3.3}$$

The ratio function plays a similar role as the importance weight in finite horizon problems. While the classic importance weight only corrects the distribution of actions between behavior policy and target policy, the ratio here also involves the correction of the states' distribution. The ratio function is connected with the average reward by

$$\eta^{\pi} = \mathbb{E}\left\{\frac{1}{T}\sum_{t=1}^{T}\omega^{\pi}(S_t,A_t)R_{t+1}\right\}$$

for any fixed trajectory length, T. An important property of ω^{π} is that for any state-action function $f(s, a)$ (not only Q^{π}),

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\omega^{\pi}(S_t,A_t)\left\{f(S_t,A_t) - \sum_{a'}\pi(a'\,|\,S_{t+1})f(S_{t+1},a')\right\}\right] = 0. \tag{3.4}$$

This orthogonality is key to develop the estimator for ω^{π} (see Section 4.3).

3.2. Efficient influence function.

In this subsection, we derive the EIF of η^{π} for a fixed policy π under time-homogeneous Markov Decision Process described in Section 2. Recall that the semiparametric efficiency bound is the supremum of the Cramèr-Rao bounds for all parametric submodels (Newey, 1990). EIF is defined as the influence function of a regular estimator that achieves the semiparametric efficiency bound. For more details, refer to Bickel et al. (1993) and Van der Vaart (2000). The EIF of η^{π} is given by the following theorem. The proof is provided in Appendix A.

Theorem 3.1.—Suppose the states in the trajectory, D, evolve according to the time-homogeneous Markov process and Assumption 2 holds. Consider a policy, π, such that Assumption 1 holds. Then the EIF of the average reward, η^{π}, is

$$\phi^{\pi}(D) = \frac{1}{T}\sum_{t=1}^{T}\omega^{\pi}(S_t,A_t)\{R_{t+1} + U^{\pi}(S_t,A_t,S_{t+1}) - \eta^{\pi}\}$$

where

$$U^{\pi}(s,a,s') := \sum_{a'}\pi(a'\,|\,s')Q^{\pi}(s',a') - Q^{\pi}(s,a)\,. \tag{3.5}$$

Remark 1.—Recall that we impose the Markovian and time-stationary assumptions on the data-generating process. Even though the parameter of interest here (i.e. the average reward η^{π}) is one-dimensional, there may exist multiple, non-efficient influence functions as a result of these assumptions on the multivariate distribution.

3.3. Doubly robust estimator.

It is known that EIF can be used to derive a semiparametric estimator (see, for example, Chap. 25 in Van der Vaart (2000)). We follow this approach. Specifically, suppose $\widehat{U}_n^{\pi}$ and $\widehat{\omega}_n^{\pi}$ are estimators of U^{π} and ω^{π} respectively. Then we estimate η^{π} by solving for η in the plug-in estimating equation: $\mathbb{P}_n\{(1/T)\sum_{t=1}^{T}\widehat{\omega}_n^{\pi}(S_t, A_t)[R_{t+1} + \widehat{U}_n^{\pi}(S_t, A_t, S_{t+1}) - \eta]\} = 0$, where for any function of the trajectory, $f(D)$, the sample average is denoted as $\mathbb{P}_n f(D) = (1/n)\sum_{i=1}^{n} f(D_i)$. The solution, $\widehat{\eta}_n^{\pi}$, is

$$\widehat{\eta}_n^{\pi} = \frac{\mathbb{P}_n\left[(1/T)\sum_{t=1}^{T}\widehat{\omega}_n^{\pi}(S_t, A_t)\left\{R_{t+1} + \widehat{U}_n^{\pi}(S_t, A_t, S_{t+1})\right\}\right]}{\mathbb{P}_n\left\{(1/T)\sum_{t=1}^{T}\widehat{\omega}_n^{\pi}(S_t, A_t)\right\}}. \tag{3.6}$$

We have the following doubly robustness of this estimator (the proof is given in Appendix A).

Theorem 3.2.—Suppose $\widehat{U}_n^{\pi}(s, a)$ and $\widehat{\omega}_n^{\pi}(s, a)$ converge in probability to deterministic limits $\bar{U}^{\pi}(s, a)$ and $\bar{\omega}^{\pi}(s, a)$ uniformly over $\mathcal{S} \times \mathcal{A}$. If either $\bar{U}^{\pi} = U^{\pi}$ or $\bar{\omega}^{\pi} = \omega^{\pi}$, then $\widehat{\eta}_n^{\pi}$ converges to η^{π} in probability.

Remark 2.—The uniform convergence in probability can be relaxed to L_2 convergence by using uniform laws of large numbers. The doubly robustness can protect against potential model mis-specifications since we only require one of two models is correct. Moreover, the doubly robust structure can be used to relax the required rate for each of the nuisance function estimation to achieve the semiparametric efficiency bound, especially if we use sample-splitting techniques (see Section 9), as discussed in Chernozhukov et al. (2018).

4. Estimators for the Nuisance Functions.

Recall the doubly robust estimator (3.6) requires the estimation of two nuisance functions, U^{π} and ω^{π}. It turns out that although these two nuisance functions are defined from different perspectives, both nuisance functions can in fact be characterized in a similar way. Both estimators can be obtained by minimizing an objective function that involves a minimizer of another objective function ("coupled estimation"). This can be viewed as a generalization of the classical M-estimator with a "plug-in estimator" in the sense that the the second objective function also involves the unknown parameters to be estimated. The idea of coupled estimation was previously used by Antos, Szepesvári and Munos (2008a); Farahmand et al. (2016) to estimate the value function in the discounted reward setting and recently by Liao, Klasnja and Murphy (2019) in the average reward setting. In what follows we provide a general coupled estimation framework and discuss the motivation for using it.

We then review the coupled estimator for relative value function and ratio function in Liao, Klasnja and Murphy (2019).

4.1. Review of coupled estimation.

Consider a setting where the true parameter (or function), θ^*, can be characterized as the minimizer of the following objective function:

$$\theta^* = \operatorname{argmin}_\theta J(\theta) = \mathbb{E}\{(l_1 \circ f_\theta)(Z)\} \tag{4.1}$$

where $l_1 : \mathbb{R} \to \mathbb{R}^+$ is a loss function composite with f (e.g., the squared loss, $l_1(x) = x^2$ and the linear model, $f_\theta(Z) = Y - \theta^\top X$, where $Z = (X, Y)$). If we can directly evaluate $f_\theta(Z)$, then we can estimate θ^* by the classical M-estimator, $\operatorname{argmin}_\theta \mathbb{P}_n\{(l_1 \circ f_\theta)(Z)\}$.

In our setting f_θ is of the form $f_\theta(Z) = \mathbb{E}[F_\theta(Z') \mid Z]$ and $f_\theta(Z)$ cannot be directly evaluated because we don't have an explicit formula for the conditional expectation $\mathbb{E}[F_\theta(Z') \mid Z]$. A natural idea to remedy this is to replace the unknown $f_\theta(Z)$ by $F_\theta(Z')$ and estimate θ^* by $\operatorname{argmin}_\theta \mathbb{P}_n\{(l_1 \circ F_\theta(Z'))\}$. Unfortunately this estimator is biased in general. To see this, suppose $l_1(x) = x^2$. We note that the limit of the new objective function, $\mathbb{P}_n\{(l_1 \circ F_\theta(Z'))\}$, is then $\tilde{J}(\theta) = \mathbb{E}\{(l_1 \circ F_\theta(Z'))\} = J(\theta) + \Delta(\theta)$ where $\Delta(\theta) = \mathbb{E}[\operatorname{Var}\{F_\theta(Z') \mid Z\}]$. The minimizer of $\tilde{J}(\theta)$ is not necessarily θ^* unless further conditions are imposed (e.g., $\operatorname{Var}\{F_\theta(Z') \mid Z\}$ is independent of θ, which is often not the case in our setting).

The high level idea of coupled estimation is to first estimate f_θ for each θ, denoted by $\hat{f}_\theta$, and then estimate θ^* by the plug-in estimator, $\operatorname{argmin}_\theta \mathbb{P}_n\{(l_1 \circ \hat{f}_\theta)(Z)\}$. A standard empirical risk minimization can be applied to obtain a consistent estimator for f_θ, e.g., $\hat{f}_\theta = \operatorname{argmin}_{g \in \mathcal{G}} \mathbb{P}_n[l_2\{(F_\theta(Z'), g(Z)\}]$ for some loss function $l_2 : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$ and a function space, $\mathcal{G}$ to approximate f_θ. We call the estimator coupled because the objective function (i.e., $\mathbb{P}_n\{(l_1 \circ \hat{f}_\theta)(Z)\}$) involves $\hat{f}_\theta$ which itself is an minimizer of another objective function (i.e., $\mathbb{P}_n l_2((F_\theta(Z'), g(Z))$) for each θ.

4.2. Relative value function estimator.

Recall the doubly robust estimator requires an estimator of U^π. It is enough to learn one specific version of Q^π. More specifically, define a shifted value function by $\tilde{Q}^\pi(s, a) = Q^\pi(s, a) - Q^\pi(s^*, a^*)$ for some specific state-action pair (s^*, a^*). By restricting to $Q(s^*, a^*) = 0$, the solution of Bellman equations (3.2) is unique and given by $\{\eta^\pi, \tilde{Q}^\pi\}$. Below we derive a coupled estimator for the shifted value function, $\tilde{Q}^\pi$, using the coupled estimation framework in Section 4.1.

Let $Z_t = (S_t, A_t, S_{t+1})$ be the transition sample at time t. For a given (η, Q) pair, let

$$\delta^\pi(Z_t; \eta, Q) = R_{t+1} + \sum_{a'} \pi(a' \mid S_{t+1}) Q(S_{t+1}, a) - Q(S_t, A_t) - \eta \tag{4.2}$$

be the so-called temporal difference (TD) error. The Bellman equation then becomes $\mathbb{E}[\delta^\pi(Z_t; \eta, Q) \mid S_t = s, A_t = a] = 0$ for all state-action pair, (s, a). As a result, we have

$$\{\eta^\pi, \widetilde{Q}^\pi\} \in \underset{\eta, Q}{\operatorname{argmin}}\, \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\left(\mathbb{E}[\delta^\pi(Z_t;\eta,Q)\,|\,S_t, A_t]\right)^2\right].$$

Note that above we choose the squared loss for simplicity; a general loss function can also be applied. We see that it fits in the coupled estimation framework presented in the previous section. In particular, $\theta^* = \{\eta^\pi, \widetilde{Q}^\pi\}$ and f_θ becomes the Bellman error, i.e., $\mathbb{E}[\delta^\pi(Z_t;\eta,Q)\,|\,S_t = \cdot\,, A_t = \cdot\,]$. The above characterization involves the average reward, η^π. Thus in the process of obtaining an estimator of the relative value function, we will also obtain an estimator of the average reward. See the end of this subsection for discussion.

We use $\mathcal{F}$ and $\mathcal{G}$ to denote two classes of functions of state-action. We use $\mathcal{F}$ to model the shifted value function, $\widetilde{Q}^\pi$, and thus require $f(s^*, a^*) = 0$ for all $f \in \mathcal{F}$. We use $\mathcal{G}$ to approximate the conditional mean of the Bellman error. In addition, $J_1\colon \mathcal{F} \to \mathbb{R}^+$ and $J_2\colon \mathcal{G} \to \mathbb{R}^+$ are two regularizers that measure the complexities of these two functional classes respectively. Given tuning parameters (λ_n, μ_n), the coupled estimator, denoted by $(\widehat{\eta}_n^\pi, \widehat{Q}_n^\pi)$, is obtained by solving

$$(\widetilde{\eta}_n^\pi, \widehat{Q}_n^\pi) = \underset{(\eta, Q) \in \mathbb{R} \times \mathcal{F}}{\operatorname{argmin}}\, \mathbb{P}_n\left[\frac{1}{T}\sum_{t=1}^{T}\widehat{g}_n^\pi(S_t, A_t; \eta, Q)^2\right] + \lambda_n J_1^2(Q), \tag{4.3}$$

where $\widehat{g}_n^\pi(\cdot\,, \cdot\,; \eta, Q)$ is the projected Bellman error at (η, Q):

$$\widehat{g}_n^\pi(\cdot\,, \cdot\,; \eta, Q) = \underset{g \in \mathcal{G}}{\operatorname{argmin}}\, \mathbb{P}_n\left[\frac{1}{T}\sum_{t=1}^{T}\left(\delta^\pi(Z_t;\eta,Q) - g(S_t, A_t)\right)^2\right] + \mu_n J_2^2(g)\,. \tag{4.4}$$

Given the estimator of the (shifted) relative value function, $\widehat{Q}_n^\pi$, we form the estimator of U^π by $\widehat{U}_n^\pi(s, a, s') = \sum_{a'} \pi(a' \mid s')\widehat{Q}_n^\pi(s', a') - \widehat{Q}_n^\pi(s, a)$.

Throughout this paper, we use tuning parameters, (λ_n, μ_n), that do not depend on the policy. In the setting where the policy class is highly complex and the corresponding relative value functions are very different, it could be beneficial to select the tuning parameters locally at a cost of higher computation burden.

Recall that the goal here is to estimate relative value function, U^π, and then plug $\widehat{U}_n^\pi$ in the doubly robust estimator (3.6). The above $\widetilde{\eta}_n^\pi$ is only used to help estimate the relative function. In fact, Liao, Klasnja and Murphy (2019) proposed using $\widetilde{\eta}_n^\pi$ to estimate the average reward. The advantage of our doubly robust estimator (3.6), $\widehat{\eta}_n^\pi$, over $\widetilde{\eta}_n^\pi$ is that the consistency of $\widehat{\eta}_n^\pi$ is guaranteed as long as at least one of the nuisance function is estimated consistently (Theorem 3.2).

4.3. Ratio function estimator.

Below we derive the estimator for the ratio function, ω^π using the coupled estimation framework. In particular we estimate a scaled version of the ratio function (denoted by e^π

below) and then convert this back to an estimator of ω^{π}. To estimate e^{π}, we first construct a new MDP and estimate the relative value function for this new MDP (denoted by H^{π}) using the coupled estimation framework. The estimator of e^{π} is then derived from the estimator of H^{π}.

We start with introducing e^{π}:

$$e^{\pi}(s,a)=\frac{\omega^{\pi}(s,a)}{\sum_{\tilde{s},\tilde{a}}\omega^{\pi}(\tilde{s},\tilde{a})d^{\pi}(\tilde{s})\pi(\tilde{a}\,|\,\tilde{s})}. \tag{4.5}$$

By definition, $\sum_{s,a}e^{\pi}(s,a)d^{\pi}(s)\pi(a\,|\,s)=1$. If we were to replace the reward function in our MDP by $1-e^{\pi}(s,a)$, then the "average reward" of π in this new MDP is constant and equal to zero under Assumption 1 (i.e., $\sum_{s,a}\{1-e^{\pi}(s,a)\}d^{\pi}(s)\pi(a\,|\,s)=0$). The "relative value function" of policy π under the new MDP is,

$$H^{\pi}(s,a)=\lim_{t^*\to\infty}\frac{1}{t^*}\sum_{t=1}^{t^*}\mathbb{E}_{\pi}\left[\sum_{k=1}^{t}\{1-e^{\pi}(S_k,A_k)\}\,|\,S_1=s,A_1=a\right]. \tag{4.6}$$

Note that H^{π} is well-defined under Assumption 1. Furthermore, consider the following Bellman equation for the new MDP:

$$\mathbb{E}_{\pi}\{1-e^{\pi}(S_t,A_t)+H(S_{t+1},A_{t+1})\,|\,S_t=s,A_t=a\}=H(s,a). \tag{4.7}$$

Note that since the "average reward" in the new MDP is zero, the above equation only involves H. The set of solutions of (4.7) can be shown to be $\{H: H=H^{\pi}+c\mathbf{1}, c\in\mathbb{R}\}$.

Below we construct a coupled estimator for a shifted version of H^{π}, i.e., $\widetilde{H}^{\pi}=H^{\pi}-H^{\pi}(s^*,a^*)$. Recall $Z_t=(S_t, A_t, S_{t+1}, R_{t+1})$ is the transition sample at time t. For a given state-action function, H, let $\Delta^{\pi}(Z_t;H)=1-H(S_t,A_t)+\sum_{a'}\pi(a'\,|\,S_{t+1})H(S_{t+1},a')$. As a result of the above Bellman-like equation and the orthogonality property (3.4), we know that

$$\widetilde{H}^{\pi}\in\operatorname*{argmin}_{H}\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}(\mathbb{E}[\Delta^{\pi}(Z_t;H)\,|\,S_t,A_t])^2\right].$$

Now it can be seen that the estimation of $\widetilde{H}^{\pi}$ fits into the coupled estimation framework (4.1), i.e., $\theta^*=\widetilde{H}^{\pi}$ and f_{θ} is $\mathbb{E}[\Delta^{\pi}(Z_t;H)\,|\,S_t=\cdot, A_t=\cdot]$. With a slight abuse of notation, we use $\mathcal{F}$ to approximate $\widetilde{H}^{\pi}$ and $\mathcal{G}$ to form the approximation of $\mathbb{E}[\Delta^{\pi}(Z_t;H)\,|\,S_t=\cdot, A_t=\cdot]$. The coupled estimator, $\widehat{H}_n^{\pi}$, is then found by solving

$$\widehat{H}_n^{\pi}=\operatorname*{argmin}_{H\in\mathcal{F}}\mathbb{P}_n\left[\frac{1}{T}\sum_{t=1}^{T}\widehat{g}_n^{\pi}(S_t,A_t;H)^2\right]+\lambda_n' J_1^2(H), \tag{4.8}$$

where for any $H\in\mathcal{F}$, $\widehat{g}_n^{\pi}(\cdot,\cdot;H)$ solves

$$\hat{g}_n^{\pi}(\cdot,\cdot;H) = \underset{g \in \mathcal{G}}{\operatorname{argmin}}\, \mathbb{P}_n\left[\frac{1}{T}\sum_{t=1}^{T}\left(\Delta^{\pi}(Z_t;H) - g(S_t,A_t)\right)^2\right] + \mu_n' J_2^2(g)\,. \tag{4.9}$$

Recall that e^{π} can be written in terms of H^{π} by (4.7); that is, re-arranging terms,

$$\mathbb{E}_{\pi}\{1 - H(S_t,A_t) + H(S_{t+1},A_{t+1}) \mid S_t = s, A_t = a\} = e^{\pi}(s,a)\,.$$

Thus given the estimator, $\widehat{H}_n^{\pi}$, we estimate e^{π} by $\hat{e}_n^{\pi}(s,a) = \hat{g}_n(s,a;\widehat{H}_n^{\pi})$. By the definition of ω^{π}, we have $\mathbb{E}[(1/T)\sum_{t=1}^{T}\omega^{\pi}(S_t,A_t)] = 1$. Since e^{π} is a scaled version of ω^{π} up to a constant, we finally construct the estimator for ratio, ω^{π}, by scaling $\hat{e}^{\pi}$, that is,

$$\widehat{\omega}_n^{\pi}(s,a) = \hat{e}_n^{\pi}(s,a)/\mathbb{P}_n[(1/T)\sum_{t=1}^{T}\hat{e}_n^{\pi}(S_t,A_t)],\ \forall (s,a) \in \mathcal{S} \times \mathcal{A}\ . \tag{4.10}$$

Remark 3.—The above ratio function estimator was developed by Liao, Klasnja and Murphy (2019). In this paper we, for the first time, derive a finite-sample error bound for this ratio function estimator, uniformly over the policy class (Theorem B.2 in the appendix). This is the key element in establishing the finite-sample bound regret bound for the estimated optimal policy.

Remark 4.—Our ratio function estimator is different from most in the existing literature, such as Liu et al. (2018); Uehara and Jiang (2019); Nachum et al. (2019); Zhang et al. (2020), which are obtained by min-max based estimating methods. For example, Liu et al. (2018) aimed to estimate the ratio between stationary distribution induced by a known, Markovian time-stationary behavior policy and target policy, which is then used to estimate the average reward of a given policy. This is not suitable for the setting where the behavior policy is history dependent. Uehara and Jiang (2019) estimated the ratio, $\omega^{\pi}(s,a)$, based on the observation that for every state-action function f,

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\left(\omega^{\pi}(S_t,A_t)\sum_{a'}\pi(a' \mid S_{t+1})f(S_{t+1},a) - \omega^{\pi}(S_t,A_t)f(S_t,A_t)\right)\right] = 0,$$

with the restriction that $\mathbb{E}[\frac{1}{T}\sum_{t=1}^{T}\omega^{\pi}(S_t,A_t)] = 1$. Then they constructed their estimator by solving the empirical version of the following min-max optimization problem:

$$\min_{w \in \Delta}\max_{f \in \mathcal{F}'}\ \mathbb{E}^2\left[\frac{1}{T}\sum_{t=1}^{T}\left(\omega(S_t,A_t)\sum_{a'}\pi(a' \mid S_{t+1})f(S_{t+1},a) - \omega(S_t,A_t)f(S_t,A_t)\right)\right],$$

where Δ is a simplex space and $\mathcal{F}'$ is a set of discriminator functions. This method minimizes the upper bound of the bias of their average reward estimator if the state-action value function is contained in $\mathcal{F}'$. They proved consistency of their ratio and average reward

estimators in the parametric setting, that is, where $\omega^{\pi}(S_t, A_t)$ can be modelled parametrically and $\mathscr{F}$ is a finite dimensional space. Subsequently Zhang et al. (2020) developed a general min-max based estimator by considering variational f-divergence, which subsumes the case in Uehara and Jiang (2019). Unfortunately, there are no error bounds guarantee for ratio function estimators developed in Uehara and Jiang (2019) and Zhang et al. (2020). Our ratio estimator appears closely related to the estimation developed by Nachum et al. (2019) as they also formulated the ratio estimator as a minimizer of a loss function. However, relying on the Fenchel's duality theorem, they still use the min-max based method to estimate the ratio. Furthermore, their method cannot be applied in average reward settings. Instead of using min-max based estimators, we use coupled estimation. This will facilitate the derivation of estimation error bounds as will be seen below. We will derive the estimation error of the ratio function, which will enable us to provide a strong theoretical guarantee, and finally demonstrate the efficiency of our average reward estimator without imposing restrictive parametric assumptions on the nuisance function estimations, see Section 5 below.

5. Theoretical Results.

5.1. Regret bound.

In this section, we provide a finite sample bound on the regret of $\hat{\pi}_n$ defined in (2.4), i.e., the difference between the optimal average reward in the policy class, Π, and the average reward of the estimated policy, $\hat{\pi}_n$.

Consider a state-action function, $f(s, a)$. Let $\mathscr{I}$ be the identity operator, i.e., $\mathscr{I}(f) = f$. Denote the conditional expectation operator by $\mathscr{P}^{\pi} f : (s, a) \mapsto \mathbb{E}_{\pi}[f(S_{t+1}, A_{t+1}) | S_t = s, A_t = a]$. Let the expectation under stationary distribution induced by π be $\mu^{\pi}(f) = \int f(s, a)\, \mu^{\pi}(ds, da)$. Denote by $\|\cdot\|_{\text{TV}}$ the total variation distance between two probability measures. For a function $g(s, a, s')$, define $\|g\|^2 = \mathbb{E}\{(1/T)\sum_{t=1}^{T} g^2(S_t, A_t, S_{t+1})\}$. For a set $\mathscr{X}$ and $M > 0$, let $\mathscr{B}(X, M)$ be the class of bounded functions on $\mathscr{X}$ such that $\|f\|_{\infty} \leq M$. Denote by $N(\epsilon, \mathscr{F}, \|\cdot\|)$ the ϵ-covering number of a set of functions, $\mathscr{F}$, with respect to the norm, $\|\cdot\|$.

We make use of the following assumption on Π.

Assumption 3.—The policy class, $\Pi = \{\pi_\theta : \theta \in \Theta \subset \mathbb{R}^p\}$, satisfies:

(3–1) $\Theta \subset \mathbb{R}^p$ is compact and let $\text{diam}(\Theta) = \sup_{\theta_1, \theta_2 \in \Theta} \|\theta_1 - \theta_2\|_2$.

(3–2) There exists $L_\Theta > 0$, such that for $\theta_1, \theta_2 \in \Theta$ and for all $(s, a) \in \mathscr{S} \times \mathscr{A}$, the following holds

$$|\pi_{\theta_1}(a|s) - \pi_{\theta_2}(a|s)| \leq L_\Theta \|\theta_1 - \theta_2\|_2.$$

(3–3) There exists constants $C_0 > 0$ and $0 \leq \beta < 1$, such that for every $\pi \in \Pi$, the following hold for all $t \geq 1$:

$$\|P^{\pi}(S_t = \cdot | S_1 = s) - d^{\pi}(\cdot)\|_{\text{TV}} \leq C_0 \beta^t, \tag{5.1}$$

$$\left\| (\mathscr{P}^{\pi})^{t} f - \mu^{\pi}(f) \right\| \leq C_0 \| f \| \beta^{t}. \tag{5.2}$$

Remark 5.—The Lipschitz property of the policy class (3–2) is used to control the complexity of nuisance function induced by Π, that is, $\{U^{\pi}(\cdot,\cdot,\cdot) : \pi \in \Pi\}$ and $\{\omega^{\pi}(\cdot,\cdot) : \pi \in \Pi\}$. This is commonly assumed in the finite-time horizon problems (e.g., Zhou et al. (2017)). Assumptions (3–1) and (3–2) can be easily satisfied by many policy classes such the one we used in Section 7. Our analysis can be extended to more general policy classes if a similar complexity property holds for these two nuisance function classes. Intuitively the constant β in the assumption (3–3) relates to the "mixing time" of the Markov chain induced by $\pi \in \Pi$. A similar assumption was used by Van Roy (1998); Liao, Klasnja and Murphy (2019) in average reward setting. Specifically, Equation (5.1) in Assumption (3–3) is used to show that two nuisance functions U^{π} and ω^{π} are Lipschitz continuous with respect to the policy parameter θ so that we can quantify their estimation error uniformly over the policy class. See Lemma C.1 of Supplementary Material for more details. Equation (5.2) in Assumption (3–3) basically requires an exponential convergence rate of the policy induced Markov chain to the stationary distribution in terms of the expectation under the L_2-norm with respect to the data generating process. This assumption, together with Assumption (5–3) stated below is used to guarantee the Bellman operator for U^{π} based on Equation (3.2) (or a similar quantity related to the ratio function estimation defined in Lemma B.4 of Supplementary Material) is well-posed in the sense of L_2-norm with respect to the data generating process so as to derive their estimation errors. See Lemma B.5 of Liao, Klasnja and Murphy (2019) and Lemma B.4 of Supplementary Material for more details.

Recall that we use the same pair of function classes $(\mathscr{F}, \mathscr{G})$ in the coupled estimation for both U^{π} and ω^{π}. We make the following assumptions on $(\mathscr{F}, \mathscr{G})$.

Assumption 4.—The function classes, $(\mathscr{F}, \mathscr{G})$, satisfy the following:

- **(4–1)** $\mathscr{F} \subset \mathscr{B}(\mathcal{S} \times \mathcal{A}, F_{\max})$ and $\mathscr{G} \subset \mathscr{B}(\mathcal{S} \times \mathcal{A}, G_{\max})$ (4–2) $f(s^*, a^*) = 0$, $f \in \mathscr{F}$.
- **(4–3)** The regularization functionals, J_1 and J_2, are pseudo norms and induced by the inner products $J_1(\cdot,\cdot)$ and $J_2(\cdot,\cdot)$, respectively.
- **(4–4)** Let $\mathscr{F}_M = \{f \in \mathscr{F} : J_1(f) \leq M\}$ and $\mathscr{G}_M = \{g \in \mathscr{G} : J_2(g) \leq M\}$. There exists C_1 and $\alpha \in (0, 1)$ such that for any $\epsilon, M > 0$,

$$\max\{\log N(\epsilon, \mathscr{G}_M, \|\cdot\|_{\infty}), \log N(\epsilon, \mathscr{F}_M, \|\cdot\|_{\infty})\} \leq C_1 \left(\frac{M}{\epsilon}\right)^{2\alpha}$$

Remark 6.—The boundedness assumption on $\mathscr{F}$ and $\mathscr{G}$ are used to simplify the analysis and can be relaxed by truncating the estimators. We restrict $f(s^*, a^*) = 0$ for all $f \in \mathscr{F}$ because $\mathscr{F}$ is used to model $\widetilde{Q}^{\pi}$ and $\widetilde{H}^{\pi}$, which by definition satisfies $\widetilde{Q}^{\pi}(s^*, a^*) = 0$ and $\widetilde{H}^{\pi}(s^*, a^*) = 0$. In Section 6, we show how to shape an arbitrary kernel function to ensure this is satisfied automatically when $\mathscr{F}$ is RKHS. The complexity assumption (4–4) on $\mathscr{F}$ and $\mathscr{G}$ are satisfied for common function classes, for example RKHS and Sobolev spaces

(Steinwart and Christmann, 2008; Györfi et al., 2006). Taking the Sobolev spaces as an example, the entropy exponent α will be $p/\tilde{q}$, where p is the dimension of state-variables and $\tilde{q}$ is the number of continuous derivatives possessed by the functions in the corresponding space. Assumption (4–4) is imposed to control the estimation error for two nuisance functions.

We now introduce the assumption that is used to bound the estimation error of value function uniformly over the policy class. Define the projected Bellman error operator:

$$g_\pi^*(\cdot,\cdot;\eta,Q) := \underset{g\in\mathcal{G}}{\operatorname{argmin}}\,\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\{\delta^\pi(Z_t;\eta,Q)-g(S_t,A_t)\}^2\right]$$

where δ^π is given in (4.2).

Assumption 5.—The triplet, $(\Pi, \mathcal{F}, \mathcal{G})$, satisfies the following:

- **(5–1)** $\widetilde{Q}^\pi(\cdot,\cdot)\in\mathcal{F}$ for $\pi\in\Pi$ and $\sup_{\pi\in\Pi}J_1(\widetilde{Q}^\pi)<\infty$.
- **(5–2)** $0\in\mathcal{G}$.
- **(5–3)** There exits $\kappa>0$, such that $\inf\{\|g_\pi^*(\cdot,\cdot;\eta,Q)\|:\|\mathbb{E}[\delta^\pi(Z_t;\eta,Q)|S_t=\cdot,A_t=\cdot]\|=1,\ |\eta|\le R_{\max},Q\in\mathcal{F},\pi\in\Pi.\}\ge\kappa$
- **(5–4)** There exists two constants C_2, C_3 such that $J_2\{g_\pi^*(\cdot,\cdot;\eta,Q)\}\le C_2+C_3J_1(Q)$ holds for all $\eta\in\mathbb{R}$, $Q\in\mathcal{F}$ and $\pi\in\Pi$.

Remark 7.—Assumption (5–1) basically assumes that the non-parametric function class $\mathcal{F}$ can model $\widetilde{Q}^\pi$ correctly, which is mild. Note that in the coupled estimator of $\widetilde{Q}^\pi$, we do not require the much stronger condition that the Bellman error for every tuple of (η, Q, π) is correctly modeled by $\mathcal{G}$. In other words, $\mathbb{E}[\delta^\pi(Z_t;\eta,Q)|S_t=\cdot,A_t=\cdot]$ is not required to belong to $\mathcal{G}$. Instead, the combination of conditions (5–2) and (5–3) is enough to guarantee the consistency of the coupled estimator (recall that the Bellman error is zero at $\{\eta^\pi,\widetilde{Q}^\pi\}$). The last condition (5–4) essentially requires the transition matrix is sufficiently smooth so that the complexity of the projected Bellman error, $J_2\{g_\pi^*(\cdot,\cdot;\eta,Q)\}$, can be controlled by $J_1(Q)$, the complexity of Q (see Farahmand et al. (2016) for an example).

A similar set of conditions are employed to bound the estimation of ratio function. For $\pi\in\Pi$ and $H\in\mathcal{F}$, define the projected error:

$$g_\pi^*(\cdot,\cdot;H)=\underset{g\in\mathcal{G}}{\operatorname{argmin}}\,\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\{\Delta^\pi(Z_t;H)-g(S_t,A_t)\}^2\right]$$

where, as before, $\Delta^\pi(Z_t;H)=1-H(S_t,A_t)+\sum_{a'}\pi(a'|S_{t+1})H(S_{t+1},a')$.

Assumption 6.—The triplet, $(\Pi, \mathcal{F}, \mathcal{G})$, satisfies the following:

(6–1) For $\pi \in \Pi$, $\widetilde{H}^{\pi}(\cdot, \cdot) \in \mathscr{F}$, and $\sup_{\pi \in \Pi} J_1(\widetilde{H}^{\pi}) < \infty$.

(6–2) $e^{\pi}(\cdot, \cdot) \in \mathscr{G}$, for $\pi \in \Pi$.

(6–3) There exits $\kappa' > 0$, such that
$\inf\{g_{\pi}^{*}(\cdot, \cdot; H) - g_{\pi}^{*}(\cdot, \cdot; \widetilde{H}^{\pi}) : (\mathscr{I} - \mathscr{P}^{\pi})(H - \widetilde{H}^{\pi}) = \}1, H \in \mathscr{F}, \pi \in \Pi\} \geq \kappa'$.

(6–4) There exists two constants C_2', C_3' such that $J_2\{g_{\pi}^{*}(\cdot, \cdot; H)\} \leq C_2' + C_3' J_1(H)$ holds for $H \in \mathscr{F}$ and $\pi \in \Pi$.

Remark 8.—The interpretation of Assumption 6 is similar to that of Assumption 5. Specifically, Assumption (6–1) basically assumes the non-parametric function class $\mathscr{F}$ can model $\widetilde{H}^{\pi}$ correctly. As in the case of estimation of relative value function, we do not require the correct modelling of $\mathbb{E}[\Delta^{\pi}(Z_t; H) | S_t = \cdot, A_t = \cdot]$ for every $(\pi, H) \in \Pi \times \mathscr{F}$ but instead only assume Assumptions (6–2) and (6–3) hold. A major difference between Assumption 5 and 6 is that (5–2) is now replaced by (6–2). This is because according to the Bellman-like equation (4.7), we have $\mathbb{E}[\Delta^{\pi}(Z_t; \widetilde{H}^{\pi}) | S_t = s, A_t = a] = e^{\pi}(s, a)$.

Theorem 5.1.—Suppose Assumptions 1 to 6 hold. Let $\hat{\pi}_n$ be the estimated policy (2.3) in which the nuisance functions are estimated with tuning parameters $\mu_n = \lambda_n = \mu_n' = \lambda_n' = Ln^{-1/(1+\alpha)}$, for some constant L > 0. Define $\beta_k = \frac{1}{1+\alpha}\{1 - (1-\alpha)2^{-k+1}\}$. Fix any integer k ≥ 2, δ ∈ (0, 1) and sufficiently large n. With probability at least 1 − δ, we have

$$\text{Regret}(\hat{\pi}_n) \leq C(\delta)\left(p^{1/2} n^{-1/2} + p n^{-\beta_k}\right),$$

where C(δ) is a function of k, L, $F_{\max}$, $G_{\max}$, L_{Θ}, diam(Θ), $\sup_{\pi \in \Pi} J_1(\widetilde{H}^{\pi})$, $\sup_{\pi \in \Pi} J_1(\widetilde{Q}^{\pi})$, $\sup_{\pi \in \Pi} \|\omega^{\pi}\|$, α, constants $\{C_0, C_1, C_2, C_3, C_2', C_3'\}$, κ, κ′, β, $p_{\min}$ and $\|\frac{d_{T+1}}{d_D}\|_{\infty}$.

Remark 9.—Recall that p is the number of parameters in the policy, α is given in (4–4), and n is the number of trajectories in the data. Theorem 5.1 shows that when the tuning parameters are of the order $O(n^{-1/(1+\alpha)})$, the regret of the estimated policy is $O\left(p^{1/2} n^{-1/2} + p n^{-\beta_k}\right)$. The leading term (in terms of n), $O(\sqrt{p/n})$, corresponds to the regret of an estimated policy as if the nuisance functions are known beforehand. The second term is due to the estimation error of nuisance functions. In particular, we show in Theorem B.1 in Section B of the appendix that the uniform estimation error of the relative value function is of $O(pn^{-1/(1+\alpha)})$ and in Theorem B.2 in the same section that the uniform estimation error of ratio is of $O\left(pn^{-\beta_k}\right)$ (see the remark after Theorem B.2 for why the rate depends on k). Note that the error of ratio is the dominant term as $\beta_k < 1/(1+\alpha)$ and β_k can be chosen arbitrarily close to $\frac{1}{1+\alpha}$ by choosing a sufficiently large k. Therefore the proposed ratio estimator can achieve the near-optimal nonparametric convergence rate. See the proof of Theorem B.2 in Section B.1 of the appendix for more details. To the best of our knowledge, this is the first result that characterizes the regret of the estimated optimal in-class policy in the infinite horizon setting.

5.2. Asymptotic results.

In this section, we prove that the average reward for our estimator of the optimal policy converges to the optimal average reward at a parametric rate (i.e., $\sqrt{n}$). Recall $\phi^{\pi}(D)$ is the efficient influence function of η^{π} given in Theorem 3.1.

Theorem 5.2.—Suppose Assumptions 1 to 6 hold. For each n ≥ 1, let $\hat{\eta}_n^{\pi}$ be the doubly robust estimator defined in (3.6) and $\hat{\pi}_n$ be the estimated policy defined in (2.3) with tuning parameters $\mu_n = \lambda_n = \mu_n^{'} = \lambda_n^{'} = Ln^{-1/(1+\alpha)}$, for some constant L > 0. Then as n → ∞,

- **i.** $\{\sqrt{n}(\hat{\eta}_n^{\pi} - \eta^{\pi}) : \pi \in \Pi\} \Rightarrow \mathbb{G}(\pi)$ in $l^{\infty}(\Pi)$ where $\mathbb{G}(\pi)$ is a zero mean Gaussian Process with covariance function $\mathscr{C} : \Pi \times \Pi \to \mathbb{R}$, $\mathscr{C}(\pi_1, \pi_2) = \mathbb{E}\{\phi^{\pi_1}(D)\phi^{\pi_2}(D)\}$.
- **ii.** $\sqrt{n}(\hat{\eta}_n^{\hat{\pi}_n} - \sup_{\pi \in \Pi}\eta^{\pi}) \Rightarrow \sup_{\pi \in \Pi_{\max}}\mathbb{G}(\pi)$, where $\mathbb{G}(\pi)$ is the Gaussian Process defined above and $\Pi_{\max} = \operatorname{argmax}_{\pi \in \Pi}\eta^{\pi}$ is the set of policies that maximize the average reward in Π.

Remark 10.—The first result shows that the estimated average reward by the doubly robust estimator reaches the semiparametric efficiency bound when we plug in the estimator for the two nuisance functions. The double robustness structure ensures that the estimation error of nuisance functions is only of lower order and does not impact the asymptotic variance of the estimated average reward. The second result shows the asymptotic of the estimated optimal value, $\hat{\eta}_n^{\hat{\pi}_n}$, converges to the maximum of the Gaussian process at the optimal policies. When there is a unique optimal policy $\pi^* = \operatorname{argmax}_{\pi \in \Pi}\eta^{\pi}$, we have $\sqrt{n}(\hat{\eta}_n^{\hat{\pi}_n} - \eta^{\pi^*})$ weakly converges to a Gaussian distribution. Estimating the limiting distribution could be challenging (especially when there exists non-unique policies) and is left for future work. Alternatively one can consider resampling-based method to construct confidence interval for $\sup_{\pi} \eta^{\pi}$ (see the recent work by Wu and Wang (2020) in single-stage problem).

6. Practical Implementation.

In this section, we describe an algorithm to estimate an in-class optimal policy based on our efficient average reward estimator $\hat{\eta}_n^{\pi}$. Without loss of generality, we consider a binary-action setting, i.e., $\mathscr{A} = \{0, 1\}$, and the following stochastic parametrized policy class Π indexed by θ:

$$\Pi = \left\{ \pi \mid \pi(1 \mid s, \theta) = \frac{\exp(s^T\theta)}{1 + \exp(s^T\theta)}, \ \| \theta \|_{\infty} \le c, \ \theta \in \mathbb{R}^p \right\},$$

for some pre-specified constant $c > 0$. Note that other link functions such as the probit function might be used here instead. Here $\| \cdot \|_{\infty}$ refers to sup-norm in Euclidean space. We fix $c = 10$ throughout our paper. In addition, we set $\mathscr{F}$ and $\mathscr{G}$ in the estimation of both value and ratio functions to be Reproducing Kernel Hilbert Spaces (RKHSs) associated with Gaussian kernels because of the representer theorem and the property of universal consistency.

The constraint on θ, $\|\theta\|_\infty \le c$, is used to maintain sufficient stochasticity in our learned policy. The stochasticity facilitates the use of $\hat{\pi}_n$ as a "warm start" policy for use by an online algorithm with future individuals. A nice side effect is that the restriction on θ provides a computational stability and can avoid degenerative cases in policy optimization similar to that when using logistic regression in classification problems (Friedman, Hastie and Tibshirani, 2001). As discussed in the introduction, we consider the simple policy class Π instead of nonparametric models such as neural networks or tree-based models mainly due to the concern of overfitting. In the batch setting, data are limited and often noisy. Using flexible function classes for modeling the policy may lead to overfitting and thus the variance of the resulting policy could be very large. The use of a simple policy class can reduce the variance while it may induce some possible bias. In addition, interpretability is critical in our batch policy learning problem. The interpretability of decision tree models are often not very stable, whereas neural networks are not very interpretable. Therefore we prefer using this simple policy class Π.

To obtain $\hat{\pi}_n \in \Pi$, we solve a multi-level optimization problem (6.1)–(6.5). Recall a multi-level optimization problem (Richardson, 1995) is a optimization problems in which the feasible set is implicitly determined by a sequence of nested optimization problems. It typically consists of an upper level optimization task that represents the objective function, and a series of (possibly nested) lower level optimization tasks that represents the feasible set.

Upper level optimization task:

$$\max_{\pi \in \Pi} \quad \frac{\mathbb{P}_n\left\{(1/T)\sum_{t=1}^{T} \widehat{\omega}_n^{\pi}(S_t, A_t)\left[R_{t+1} + \widehat{U}_n^{\pi}(S_t, A_t, S_{t+1})\right]\right\}}{\mathbb{P}_n\left\{(1/T)\sum_{t=1}^{T} \widehat{\omega}_n^{\pi}(S_t, A_t)\right\}} \tag{6.1}$$

Lower level optimization task 1:

$$(\widehat{\eta}_n^{\pi}, \widehat{Q}_n^{\pi}) = \underset{(\eta, Q) \in \mathbb{R} \times \mathcal{F}}{\operatorname{argmin}} \mathbb{P}_n\left[\frac{1}{T}\sum_{t=1}^{T}\left[\widehat{g}_n^{\pi}(S_t, A_t; \eta, Q)\right]^2\right] + \lambda_n J_1^2(Q) \tag{6.2}$$

$$\text{s.t. } \widehat{g}_n^{\pi}(\cdot, \cdot; \eta, Q) = \operatorname{argmin}_{g \in \mathcal{G}} \mathbb{P}_n\left[\frac{1}{T}\sum_{t=1}^{T}\left(\delta^{\pi}(Z_t; \eta, Q) - g(S_t, A_t)\right)^2\right] + \mu_n J_2^2(g) \tag{6.3}$$

Lower level optimization task 2:

$$\widehat{H}_n^{\pi}(\cdot, \cdot) = \underset{H \in \mathcal{F}}{\operatorname{argmin}} \mathbb{P}_n\left[\frac{1}{T}\sum_{t=1}^{T}\left[\widehat{g}_n^{\pi}(S_t, A_t; H)\right]^2\right] + \lambda_n' J_1^2(H) \tag{6.4}$$

$$\text{s.t. } \hat{g}_n^{\pi}(\cdot,\cdot;H) = \underset{g \in \mathcal{G}}{\operatorname{argmin}}\, \mathbb{P}_n\left[\frac{1}{T}\sum_{t=1}^{T}\left(\Delta^{\pi}(Z_t;H) - g(S_t,A_t)\right)^2\right] + \mu_n' J_2^2(g). \tag{6.5}$$

As a reminder, recall that in Section 4 we have defined

$$\delta^{\pi}(Z_t;\eta,Q) = R_{t+1} + \sum_{a}\pi(a'|S_{t+1})Q(S_{t+1},a) - Q(S_t,A_t) - \eta,$$

$$\hat{U}_n^{\pi}(S_t,A_t,S_{t+1}) = \sum_{a\in\mathcal{A}}\pi(a|S_{t+1})\hat{Q}_n^{\pi}(S_{t+1},a) - \hat{Q}_n^{\pi}(S_t,A_t),$$

and

$$\Delta^{\pi}(Z_t;H) = 1 - H(S_t,A_t) + \sum_{a'}\pi(a'|S_{t+1})H(S_{t+1},a').$$

Also, the ratio estimator ω_n^{π} can be obtained from $\hat{H}_n^{\pi}(\cdot,\cdot)$ by using (4.10).

The upper optimization task (6.1) is used to search for $\hat{\pi}_n$ and the two parallel lower optimization tasks (6.2)–(6.3) and (6.4)–(6.5) are used to compute two nuisance function estimators for a given $\boldsymbol{\pi} \in \Pi$, i.e., the feasible set, respectively. Note that each nuisance function estimation is itself a nested optimization sub-problem. Multi-level optimization problems in general cannot be computed by iteratively updating solutions to lower problems (6.2)–(6.3) and (6.4)–(6.5), and solutions to the upper problem (6.1), in a similar manner to coordinate descent. Hence, in order to solve this problem, one common approach is to replace the inner optimization problems (6.2)–(6.3) and (6.4)–(6.5) by their corresponding Karush-Kuhn-Tucker (KKT) conditions so that the overall problem can be equivalently formulated as a nonlinear constraint optimization problem. However, this approach can be computationally expensive and may not be suitable for large scale settings. Instead we overcome this computational obstacle by using the representer theorem and obtain the closed-form solutions for our inner optimization problems (6.2)–(6.3) and (6.4)–(6.5) respectively. After plugging these closed-form solutions into (6.1), we can use a gradient-based method to find $\hat{\pi}_n$.

6.1. RKHS reformulation.

In the following subsection, we briefly discuss how to simplify our multi-level optimization problem (6.1) using the representer theorem. The details of computation can be found in Appendix E. For the ease of illustration, we rewrite the training data $\mathcal{D}_n$ into tuples $Z_h = \{S_h, A_h, R_h, S_h'\}$ where $h = 1, \ldots, N = nT$ indexes the tuple of the transition sample in the training set $\mathcal{D}_n$, S_h and S_h' are the current and next states and R_h is the associated reward. Let $W_h = (S_h, A_h)$ be the state-action pair, and $W_h' = (S_h, A_h, S_h')$. Suppose the kernel function for the state is denoted by $k_0(s_1, s_2)$, where $s_1, s_2 \in \mathcal{S}$. In order to incorporate the action space, we can define $k((s_1,a_1),(s_2,a_2)) = \mathbb{1}_{\{a_1=a_2\}}k_0(s_1,s_2)$. Basically, we model each $Q(\cdot,a)$

separately for each arm in the RKHS with the same kernel k_0. Recall that we have to restrict the function space $\mathcal{F}$ such that $Q(s^*, a^*) = 0$ for all $Q \in \mathcal{F}$ so as to avoid the identification issue. Thus for any given kernel function k defined on $\mathcal{S} \times \mathcal{A}$, we make the following transformation by defining $k(W_h, W_j) = k_0(W_h, W_j) - k_0((s^*, a^*), W_h) - k_0((s^*, a^*), W_j) + k_0((s^*, a^*),(s^*, a^*))$ for any $1 \le h, j \le N$. One can check that the induced RKHS by $k(\cdot,\cdot)$ satisfies the constraint in $\mathcal{F}$ automatically.

We denote kernel functions for $\mathcal{F}$ and $\mathcal{G}$ by $k(\cdot,\cdot)$, $l(\cdot,\cdot)$ respectively. The corresponding inner products are defined as $\langle \cdot, \cdot \rangle_{\mathcal{F}}$ and $\langle \cdot, \cdot \rangle_{\mathcal{G}}$. We first discuss the inner minimization problem (6.2)–(6.3). Note that this is indeed a nested kernel ridge regression problem, different from the standard ridge regression. The closed form solution can be obtained as $\hat{g}_n^{\pi}(\cdot, \cdot; \eta, Q) = \sum_{h=1}^{N} l(W_h, \cdot)\hat{\gamma}(\eta, Q)$. In particular, $\hat{\gamma}(\eta, Q) = (L + \mu I_N)^{-1}\delta_N^{\pi}(\eta, Q)$, where $Q \in \mathcal{F}$ and L is the kernel matrix induced by l, $\mu = \mu_{\mathcal{D}} N$, and $\delta_N^{\pi}(\eta, Q) = (\delta^{\pi}(Z_h; \eta, Q))_{h=1}^{N}$ is a vector of TD error. Each TD error can be further written as $\delta^{\pi}(Z_h; \eta, Q) = R - \eta - \langle Q, f_{W_h'} \rangle_{\mathcal{F}}$ where

$$f_{W_h'}(\cdot) = k(W_h, \cdot) - \sum_{a'} \pi(a' \mid S_h') k((S_h', a'), \cdot) \in \mathcal{F} \ (\mathcal{S} \times \mathcal{A})$$

It can be shown that $\hat{Q}_n^{\pi}$ in (6.2) must be in the linear span $\{\sum_{h=1}^{N} \alpha_h f_{W_h'}(\cdot) : \alpha_h \in \mathbb{R},\ h = 1, \ldots, N\}$ by using the representer property.

Then we can solve the optimization problem (6.2)–(6.3). The solutions for $\{\hat{U}_n^{\pi}(W_h')\}_{h=1}^{N}$ can be found as $-\tilde{F}(\pi)\hat{\alpha}(\pi)$ where $\tilde{F}(\pi) = (\langle f_{W_h'}, f_{W_j'} \rangle_{\mathcal{F}})_{j,h=1}^{N}$ is a N by N matrix and $\hat{\alpha}(\pi)$ is the vector of coefficients with a closed-form expression (see Appendix E for details). Similarly, we can compute the closed-form solutions $\{\hat{g}_n^{\pi}(W_h, \hat{H}_n^{\pi})\}_{h=1}^{N}$ to the problem (6.4)–(6.5) as $L\hat{\nu}(\pi)$. Here $\hat{\nu}(\pi)$ is the corresponding estimated coefficients associated with the kernel matrix L. The details can be found in appendix E. Note that all of these intermediate terms except for L depends on the policy π.

Summarizing together and plugging all the intermediate results into (6.1), the multi-level optimization problem can be simplified as:

$$\max_{\pi \in \Pi} \frac{(\hat{\nu}(\pi))^{\top} L\left(R_N - \tilde{F}(\pi)\hat{\alpha}(\pi)\right)}{\hat{\nu}(\pi)^{\top} L 1_N}, \tag{6.6}$$

where 1_N is a length-N vector of all ones.

6.2. Optimization.

Note that problem (6.6) becomes a smooth nonlinear optimization with box constraints. We use limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm with box constraints (L-BFGS-B) to compute the solution $\hat{\theta}$ (Liu and Nocedal, 1989). The gradient computing is provided in appendix. The computational complexity/operations of our algorithm is of order $\Upsilon N^3 p$, where Υ is the number of iterations in our optimization algorithm. The memory requirement is of order $N^2 p$. One may implement some sub-sampling methods such as

stochastic gradient decent to further improve both computation and memory complexity of our algorithm. We will leave it for future work. Although the overall optimization problem is non-convex and, thus an optimal solution may not be achievable, the performance of our numerical experiments in the following section are quite stable and promising. Recently, there is a growing interest in studying statistical properties of algorithm-type of nonconvex M-estimators, e.g., (Mei et al., 2018; Loh et al., 2017). For many practical applications, gradient decent methods with a random initialization have been demonstrated to converge to local minima (or even global minima) that are statistically good. While this is not the focus of our paper, it will be interesting to pursue toward this direction for future research such as studying the landscape of η^{π} and its related properties.

6.3. Tuning parameters selection.

In this subsection, we discuss the choice of tuning parameters in our method. The bandwidths in the Gaussian kernels are selected using median heuristic, e.g., median of pairwise distance (Fukumizu et al., 2009). The tuning parameters (λ_n, μ_n) and (λ_n', μ_n') are selected based on 3-fold cross-validation. Given assumptions in Theorems B.1 and B.2 of the appendix that these tuning parameters are independent of the policy π, we can select them for the ratio and value functions separately. Specifically, for the tuning parameters (λ_n, μ_n) in the estimation of value function, we focus on (6.2)–(6.3). For the tuning parameters (λ_n', μ_n') in the estimation of ratio function, we focus on (6.4)–(6.5). At the first glance, one may think the selection of tuning parameters will be the same as those in the standard supervised learning. However, this actually requires an additional step as we cannot observe responses when estimating these two coupled estimators (recalled that we need to first compute projected bellman errors), in contrast to the standard kernel regression setting. In the following, we discuss our selection procedure of (λ_n, μ_n) and (λ_n', μ_n') with more details.

Algorithm 1:

Tuning parameters selection via cross-validation

Input: Data $\{Z_h\}_{h=1}^N$, a set of M policies $\{\pi_1, \cdots, \pi_M\} \subset \Pi$, a set of J candidate tuning parameters $\{(\mu_j, \lambda_j)\}_{j=1}^J$ in the value function estimation, and a set of J candidate tuning parameters $\{(\mu_j', \lambda_j')\}_{j=1}^J$ in the ratio function estimation.

Randomly split Data into K subsets: $\{Z_h\}_{h=1}^N = \{D_k\}_{k=1}^K$

Denote $e^{(1)}(m, j)$ and $e^{(2)}(m, j)$ as the total validation error for m-th policy and j-th pair of tuning parameters in value and ratio function estimation respectively, for $m = 1, \cdots M$ and $j = 1, \cdots, J$. Set their initial values as 0.

Repeat for $m = 1, \cdots, M$,

  Repeat for $k = 1, \cdots, K$,

    Repeat for $j = 1, \cdots, J$

      Use $\{Z_h\}_{h=1}^N \backslash D_k$ to compute $(\hat{\eta}_n^{\pi_m}, \hat{\alpha}(\pi_m))$ and $\hat{v}(\pi_m)$ by (6.2)–(6.3) and (6.4)–(6.5) using tuning parameters (μ_j, λ_j) and (μ_j', λ_j') respectively;

      Compute $\hat{\delta}^{\pi_m}(\cdot; \hat{\eta}(\pi_m), \hat{Q}_n^{\pi_m})$ and $\hat{\varepsilon}^{\pi_m}(\cdot; \hat{H}_n^{\pi_m})$ and their corresponding squared Bellman errors $mse^{(1)}$ and $mse^{(2)}$ on the dataset D_k by Gaussian kernel regression;

      Assign $e^{(1)}(m, j) = e^{(1)}(m, j) + mse^{(1)}$ and $e^{(2)}(m, j) = e^{(2)}(m, j) + mse^{(2)}$;

Compute $j^{(1)*} \in \text{argmin}_j \max_m e^{(1)}(m, j)$ and $j^{(2)*} \in \text{argmin}_j \max_m e^{(2)}(m, j)$

11 **Output:** $(\mu_{j^{(1)}*}^{(1)}, \lambda_{j^{(1)}*}^{(1)})$ and $(\mu_{j^{(2)}*}^{'}, \lambda_{j^{(2)}*}^{'})$.

We first randomly choose a set of candidate policies used to gauge our tuning parameters. For each candidate policy, π, in this set, we can firstly estimate $(\hat{\eta}_n^{\pi}, \hat{\alpha}(\pi))$ by the proposed method using two folds of data. Then for the value function estimation, we calculate temporal difference errors $\delta^{\pi}(\cdot; \hat{\eta}_n^{\pi}, \hat{\alpha}(\pi))$ for each transition sample in the validation set. Since we cannot observe/calculate the true bellman error, following the idea in (Farahmand and Szepesvári, 2011), we estimate the Bellman error by projecting these temporal differences on the space of $\mathcal{S} \times A$ in the validation set using the standard Gaussian kernel regression. Thus for each policy π and each pair of tuning parameters, we output the squared estimated Bellman error in the validation set as a criterion to evaluate the performance of our value function estimation. Since tuning parameters are assumed independent of policies, we then select the tuning parameters that minimize the worst case of estimated Bellman errors among the set of all candidate policies. We use the same strategy to select the tuning parameters for our ratio estimation. The details are given in the Algorithm 1. Without the independent assumptions of tuning parameters from the policies in Π, one may alternatively choose these tuning parameters jointly by maximizing $\hat{\eta}_n^{\pi}$ on the validation set, which requires large computational costs and we omit here. But it would be very interesting to study the theoretical properties of these two cross-validation procedures, or more generally, the selection of tuning parameters in the framework of couple estimation, which we leave it as future work.

7. Simulation Studies.

In this section, we consider two scenarios to evaluate the proposed algorithm. For both scenarios, we consider $S_t = (S_{t,1}, S_{t,2}, S_{t,3})$ as a three-dimensional state at each decision point t, and the action space is binary, i.e., $\mathcal{A} = \{0, 1\}$. The behavior policy used to generate actions follows Bernoulli distribution with equal probabilities. In addition, the initial state S_1 is sampled from standard multi-variate normal distribution, i.e., $S_1 \sim MVN(0, I_4)$

The first scenario we consider is a standard MDP setting. Let ξ_t follows a standard multivariate normal distribution. Then we generate the transition of states and reward functions via following models:

$$S_{t+1,1} = 0.5S_{t,1} + 2\xi_{t,1},$$

$$S_{t+1,2} = 0.25S_{t,2} + 0.125A_t + 2\xi_{t,2},$$

$$S_{t+1,3} = 0.9S_{t,3} + 0.05S_{t,3}A_t + 0.5A_t + \xi_{t,3},$$

$$R_{t+1} = 10 - 0.4S_{t,3} + 0.25S_{t,1}A_t \times (0.04 + 0.02S_{t,1} + 0.02S_{t,2}) + 0.16\xi_{t,4},$$

for $t = 1, \cdots, T$. Here $S_{t,3}$ can be interpreted as the treatment burden or fatigue.

The second scenario we consider is a non-stationary environment. In particular, we consider the same transition models as above, but let the reward function to be time-dependent. More specifically, we consider

$$R_{t+1} = 10 - \tau_t S_{t,3} + \beta_t S_{t,1} A_t (0.04 + 0.02S_{t,1} + 0.02S_{t,2}),$$

where the time-varying parameters $\beta_t = 0.25 \times \exp(-0.05(t-1))$ and $\tau_t = 0.4 \times \exp(-0.05(t-1))$. This generative model represents the scenario in which as the study progresses, the overall impact of intervention is decreasing. Note that since the reward function is non-stationary, we do not have a guarantee for our proposed algorithm to find an optimal policy.

We compare with four baseline methods, which were proposed in the setting of the discounted sum of rewards. The first two are recently proposed deep off-policy RL algorithms (Fujimoto, Meger and Precup, 2019; Kumar et al., 2019) denoted by BCQ and BEAR respectively. The underlying idea behind these two state-of-art algorithms is to conservatively estimate the optimal Q-function on the less explored state-action pair and restrict the resulting policy close to the behavior one. The third method is the celebrated fitted-Q iteration (FQI) method proposed by Ernst et al. (2005). At each iteration, relying on the optimal Bellman equation, FQI algorithm updates the estimation of the optimal Q-function via solving a supervised learning problem. The last method is V-learning proposed by (Luckett et al., 2019), which also aims to learn an optimal in-class policy. Since our goal is to maximize the long-term average reward, we set the discount factor γ in these four methods as 0.99 to approximate the average reward for comparison. In addition, to draw a relatively fair comparison, we implement these four methods using the same policy class as ours. Specifically, the first three methods will output an estimation of the optimal Q-function (defined in the discounted setting), after which we implement a weighted logistic regression to estimate the optimal in-class policy. For V-learning, we keep the default setup and use the same policy class as ours. Finally, for BEAR and BCQ, we use two-hidden layers neural networks with 32 nodes for each and ReLU activation functions to model the optimal Q-function. The other hyper-parameters are either tuned for their best performance, or recommended in the official implementation as robust choices. For FQI, we implement a kernel ridge regression at each iteration with tuning parameters selected similar to our nuisance parameter estimation.

To demonstrate the performance of our algorithm compared with other methods, we consider different combinations of the number of trajectories n and the length of each trajectory T. Specifically, we consider $(n, T) = (40, 50)$, $(40, 100)$, and $(80, 50)$. Once all estimated policies are obtained, we generate another 100 test samples with the length of trajectories 1000 using all learned policies and compute the corresponding empirical average of observed rewards. In order to compare results with the best in-class stationary policy, we combine the gradient-type optimization algorithm with Monte Carlo method to

estimate the best in-class policy that can maximize $\mathbb{E}^{\pi}\left[\frac{1}{1000}\sum_{t=1}^{1000} R_{t+1}\right]$. Specifically, for each policy with parameter θ, we generate a sample of $n = 100$ and $T = 1000$ to approximate $\mathbb{E}^{\pi}\left[\frac{1}{1000}\sum_{t=1}^{1000} R_{t+1}\right]$ by the empirical average of rewards. Then we apply L-BFGS algorithm and require θ to be between −10 to 10 to search for the best in-class stationary policy, which is treated as the oracle policy.

Results of the above two scenarios can be found in Table 1. As we can see, our algorithm performs well in finding optimal in-class stationary policies, compared with the other four baseline methods. Compared with the oracle one with the best average reward about 10, the regret of our algorithm is almost the smallest among all these methods, which is expected as we aim to maximize the average reward while the other four methods are for maximizing the discounted sum of rewards. For BEAR and BCQ with neural network models, due to the relatively small sample size and a large discount factor γ, the performances seem unstable. FQI and V-learning methods overall show competitive performances. But it can be seen that V-learning may suffer some large variance. In addition, one possible reason for the high quality performance of our method in Scenario 2 is that the time-dependent effect in reward function is exponentially decaying by our design. Therefore we expect the performance of our algorithm may not be affected severely by the non-stationarity. In addition, it can be seen that as the sample size n or the length of each trajectory T increases, the average rewards of our estimated policies are also improved, demonstrating the appealing performance of the proposed method. Finally, we remark that the maximum running time of our method for one replication in our simulation studies is less than 40 minutes.

8. Application to mobile health.

We apply the proposed method to HeartSteps. HeartSteps is mobile health application focusing on physical activity. Three studies were conducted to develop the intervention. In this work, we apply the proposed method to the data collected from the first study, which we will refer to as HS1 in the throughout. HS1 is a 42-day micro-randomized trial (Klasnja et al. (2015); Liao et al. (2016)). Each participant was provided with a Jawbone wrist tracker to collect step count data and specified five decision times, roughly 2.5 hours apart during each day, that would be good times to potentially receive contextually tailored activity suggestion message. In HS1, the activity message was sent with a fixed probability 0.6 at each of the five decision times. Our goal is to use HS1 data to learn a treatment policy that determines at each decision time whether to send the activity message (i.e., binary action).

We construct the state variable using the previous step count (the 30-min step count prior to the decision time and from yesterday), location, temperature and past notifications. We set the reward to be the log transformation of the step count in 30-min window after each decision time. In this analysis, we include 37 participants' data and exclude the decision times when participants were traveling abroad or experiencing technical issues or when the reward (i.e., post 30-min step count) was considered as missing (Klasnja et al., 2018).

Next, we construct the policy class. In this analysis, we include two state variables in the policy. The first variable is the location (home/work vs. other locations). Location is

important because people, in a more structured environment (i.e. at home or work), may respond better to an activity suggestion as compared to when they are at other locations. As a proxy for participant burden, the second variable included in the policy is "dosage", a discounted sum of the number of past activity messages sent with the discount rate chosen as 0.95. The rationale for using this variable is that receiving too many notifications in the recent past is likely to decrease the effectiveness of sending the activity message due to over-burdening participants. We consider the policy class of the form $\pi_\theta(1|s) = \text{expit}(\theta^\top \phi(s))$, $\theta \in \Theta$, where the feature vector $\phi(s) = (1, \text{dosage}, \text{location})$ and Θ is the box constraint within -10 and 10. Here dosage is standardized to be within 0 and 1.

We apply the proposed method with the tuning parameter selected by cross-validation in Algorithm 1. The estimated coefficients are [10, −10, −4.788]. Figure 1 shows the estimated policy at different combination of dosage and location. As one would expect, the learned policy tends to send fewer suggestions if the participant received many suggestions in the recent past. Also, the policy indicates that it is more effective to send the message when the user is at home/work location. The estimated average reward of this policy is 3.301. As a comparison, the estimated average reward of the simple location-based policy (i.e., send only when the user is at home/work) is 3.15 and the send-nothing policy is 2.96. Transforming to the scale of the raw step count as that in (Klasnja et al., 2018), the learned policy can result in 16% (i.e., $\exp(3.301 - 3.15) - 1 = 0.16$) improvement, which is equivalent to 40 more steps (the mean step count across all decision times in the data is 248) compared with the simple location-based policy, and 40% (i.e., $\exp(3.301 - 2.96) - 1 = 0.40$) improvement, or equivalently 101 steps more, compared with the send-nothing policy. Lastly, we remark that the running time of our real data analysis is about 2 hours, which is acceptable in the batch setting. This is ultimately different from online RL domains where the policy is usually updated upon the arrival of each observation.

9. Discussion.

Double/Debiased machine learning

An alternative way to construct the estimator for the average reward is based on the idea of double/debiased machine learning (a.k.a. cross-fitting, Bickel et al. (1993) and Chernozhukov et al. (2018)). There is growing interest in using double machine learning in causal inference and in the policy learning literature (Zhao et al., 2019) in order to relax assumptions on the convergence rates of nuisance parameters. The basic idea is to split the data into K folds. For each of the K folds, construct the estimating equation by plugging in the estimated nuisance functions that are obtained using the remaining $(K - 1)$ folds. The final estimator is obtained by solving the aggregated estimation equations. While cross-fitting requires weaker conditions on the nuisance function estimations, it indeed incurs additional computational cost, especially in our setting where nuisance functions are policy-dependent and we aim to search for the in-class optimal policy. Further, this sample splitting procedure may not be stable when the sample size is relatively small, e.g., in a typical mHealth clinical trial. A more efficient way of data splitting under the framework of MDP is needed, which we leave as future work

Computation and optimization

Our current algorithm requires relatively large computation and memory because of the non-parametric estimation and the policy-dependent structure of nuisance functions. It is therefore desirable to develop a more efficient algorithm. One possible remedy is to consider a zero-order optimization method such as Bayesian optimization (Snoek, Larochelle and Adams, 2012), which is suitable when the dimension of state variables is small. Another possible way to improve the computational efficiency is to first apply some simple algorithm to estimate a sub-optimal policy, based on which we can implement our method to estimate two nuisance parameters. Then one can develop the performance difference lemma in terms of the average reward MDP, similar to that in the discounted setting (Kakade and Langford, 2002), to construct a lower bound for $\mathscr{V}(\pi)$ using two estimated nuisance parameters. The last step is to optimize this lower bound for obtaining a better policy. This method may require less computational cost.

Tuning parameters/Model selection

In our proposed algorithm, we assume tuning parameters are independent of policies, based on which we develop a min-max cross-validation procedure for the selection of tuning parameters. Model selection in the offline RL setting, which is necessary for improving generalization of RL techniques, is often considered as a challenging task as there is no ground truth available for performance demonstration, in contrast to the online setting with simulated environment. Therefore, it will be interesting to systematically investigate how to perform model selection in offline RL and to provide theoretical guarantees.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

REFERENCES

Abounadi J, Bertsekas D and Borkar VS (2001). Learning algorithms for Markov decision processes with average cost. SIAM Journal on Control and Optimization 40 681–698.

Agarwal R, Schuurmans D and Norouzi M (2020). An optimistic perspective on offline reinforcement learning. In International Conference on Machine Learning 104–114. PMLR.

Antos A, Szepesvári C and Munos R (2008a). Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. Machine Learning 71 89–129.

Antos A, Szepesvári C and Munos R (2008b). Fitted Q-iteration in continuous action-space MDPs. In Advances in neural information processing systems 9–16.

Athey S and Wager S (2017). Efficient policy learning. arXiv preprint arXiv:1702.02896.

Bickel PJ, Klaassen CA, Bickel PJ, Ritov Y, Klaassen J, Wellner JA and Ritov Y (1993). Efficient and adaptive estimation for semiparametric models 4. Johns Hopkins University Press Baltimore.

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.

Dudík M, Erhan D, Langford J, Li L et al. (2014). Doubly robust policy evaluation and optimization. Statistical Science 29 485–511.

Ernst D, Geurts P, Wehenkel L and Littman L (2005). Tree-based batch mode reinforcement learning. Journal of Machine Learning Research 6 503–556.

Ertefaie A and Strawderman RL (2018). Constructing dynamic treatment regimes over indefinite time horizons. Biometrika 105 963–977.

Farahmand A-M and Szepesvári C (2011). Model selection in reinforcement learning. Machine learning 85 299–332.

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.

Friedman J, Hastie T and Tibshirani R (2001). The elements of statistical learning 1. Springer series in statistics New York.

Fujimoto S, Meger D and Precup D (2019). Off-policy deep reinforcement learning without exploration. In International Conference on Machine Learning 2052–2062. PMLR.

Fukumizu K, Gretton A, Lanckriet GR, Schölkopf B and Sriperumbudur BK (2009). Kernel choice and classifiability for RKHS embeddings of probability distributions. In Advances in neural information processing systems 1750–1758.

Györfi L, Kohler M, Krzyzak A and Walk H (2006). A distribution-free theory of nonparametric regression. Springer Science & Business Media.

Hernández-lerma O and Lasserre JB (1999). Further topics on discrete-time Markov control processes 42. Springer.

Jiang N and Li L (2016). Doubly robust off-policy value evaluation for reinforcement learning. In International Conference on Machine Learning 652–661. PMLR.

Kakade S and Langford J (2002). Approximately optimal approximate reinforcement learning. In In Proc. 19th International Conference on Machine Learning. Citeseer.

Kallus N and Uehara M (2019a). Double reinforcement learning for efficient off-policy evaluation in markov decision processes. arXiv preprint arXiv:1908.08526.

Kallus N and Uehara M (2019b). Efficiently breaking the curse of horizon: Double reinforcement learning in infinite-horizon processes. arXiv preprint arXiv:1909.05850.

Klasnja P, Hekler EB, Shiffman S, Boruvka A, Almirall D, Tewari A and Murphy SA (2015). Micro-randomized trials: An experimental design for developing just-in-time adaptive interventions. Health Psychology 34 1220.

Klasnja P, Smith S, Seewald NJ, Lee A, Hall K, Luers B, Hekler EB and Murphy SA (2018). Efficacy of contextually tailored suggestions for physical activity: a micro-randomized optimization trial of HeartSteps. Annals of Behavioral Medicine.

Kosorok MR and Laber EB (2019). Precision medicine. Annual review of statistics and its application 6 263–286.

Kumar A, Fu J, Soh M, Tucker G and Levine S (2019). Stabilizing off-policy q-learning via bootstrapping error reduction. Advances in Neural Information Processing Systems 32.

Laber EB, Lizotte DJ, Qian M, Pelham WE and Murphy SA (2014). Dynamic treatment regimes: Technical challenges and applications. Electronic journal of statistics 8 1225. [PubMed: 25356091]

Lagoudakis MG and Parr R (2003). Least-squares policy iteration. Journal of machine learning research 4 1107–1149.

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.

Liao P, Klasjna P, Tewari A and Murphy SA (2016). Micro-Randomized Trials in mHealth. Statistics in Medicine 35 1944–71. [PubMed: 26707831]

Liu DC and Nocedal J (1989). On the limited memory BFGS method for large scale optimization. Mathematical programming 45 503–528.

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 5356–5366.

Liu Y, Swaminathan A, Agarwal A and Brunskill E (2019). Off-Policy Policy Gradient with State Distribution Correction. arXiv preprint arXiv:1904.08473.

Loh P-L et al. (2017). Statistical consistency and asymptotic normality for high-dimensional robust *M*-estimators. The Annals of Statistics 45 866–896.

Luckett DJ, Laber EB, Kahkoska AR, Maahs DM, Mayer-Davis E and Kosorok MR (2019). Estimating dynamic treatment regimes in mobile health using V-learning. Journal of the American Statistical Association just-accepted 1–39. [PubMed: 34012183]

Mahadevan S (1996). Average reward reinforcement learning: Foundations, algorithms, and empirical results. Machine learning 22 159–195.

Mei S, Bai Y, Montanari A et al. (2018). The landscape of empirical risk for nonconvex losses. The Annals of Statistics 46 2747–2774.

Munos R and Szepesvári C (2008). Finite-time bounds for fitted value iteration. Journal of Machine Learning Research 9 815–857.

Murphy SA, van der Laan MJ, Robins JM and Group CPPR (2001). Marginal mean models for dynamic regimes. Journal of the American Statistical Association 96 1410–1423. [PubMed: 20019887]

Murphy SA, Deng Y, Laber EB, Maei HR, Sutton RS and Witkiewitz K (2016). A batch, off-policy, actor-critic algorithm for optimizing the average reward. arXiv preprint arXiv:1607.05047.

Nachum O, Chow Y, Dai B and Li L (2019). Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. In Advances in Neural Information Processing Systems 2315–2325.

Nahum-Shani I, Smith SN, Spring BJ, Collins LM, Witkiewitz K, Tewari A and Murphy SA (2016). Just-in-Time Adaptive Interventions (JITAIs) in mobile health: key components and design principles for ongoing health behavior support. Annals of Behavioral Medicine 1–17. [PubMed: 26318593]

Naik A, Shariff R, Yasui N and Sutton RS (2019). Discounted reinforcement learning is not an optimization problem. arXiv preprint arXiv:1910.02140.

Newey WK (1990). Semiparametric efficiency bounds. Journal of applied econometrics 5 99–135.

Ormoneit D and Sen S (2003). Kernel-Based Reinforcement Learning. In Machine Learning 161–178.

Precup D (2000). Eligibility traces for off-policy policy evaluation. Computer Science Department Faculty Publication Series 80.

Puterman ML (1994). Markov Decision Processes: Discrete Stochastic Dynamic Programming.

Richardson GB (1995). The theory of the market economy. Revue économique 1487–1496.

Robins JM, Rotnitzky A and Zhao LP (1994). Estimation of regression coefficients when some regressors are not always observed. Journal of the American statistical Association 89 846–866.

Sharma H, Jafarnia-Jahromi M and Jain R (2020). Approximate relative value learning for average-reward continuous state MDPs. In Uncertainty in Artificial Intelligence 956–964. PMLR.

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

Shi C, Wan R, Chernozhukov V and Song R (2021). Deeply-Debiased Off-Policy Interval Estimation. arXiv preprint arXiv:2105.04646.

Snoek J, Larochelle H and Adams RP (2012). Practical bayesian optimization of machine learning algorithms. Advances in neural information processing systems 25.

Steinwart I and Christmann A (2008). Support vector machines. Springer Science & Business Media.

Sutton RS and Barto AG (2018). Reinforcement learning: An introduction. MIT press.

Tang Z, Feng Y, Li L, Zhou D and Liu Q (2020). Doubly Robust Bias Reduction in Infinite Horizon Off-Policy Estimation. In International Conference on Learning Representations.

Thomas P and Brunskill E (2016). Data-efficient off-policy policy evaluation for reinforcement learning. In International Conference on Machine Learning 2139–2148.

Uehara M and Jiang N (2019). Minimax Weight and Q-Function Learning for Off-Policy Evaluation. arXiv preprint arXiv:1910.12809.

van der Vaart AW (2000). Asymptotic statistics 3. Cambridge university press.

van Roy B (1998). Learning and value function approximation in complex decision processes, PhD thesis, Massachusetts Institute of Technology.

Voloshin C, Le HM, Jiang N and Yue Y (2019). Empirical Study of Off-Policy Policy Evaluation for Reinforcement Learning. arXiv preprint arXiv:1911.06854.

Wan Y, Naik A and Sutton RS (2021). Learning and planning in average-reward markov decision processes. In International Conference on Machine Learning 10653–10662. PMLR.

Wu Y and Wang L (2020). Resampling-based confidence intervals for model-free robust inference on optimal treatment regimes. Biometrics n/a.

Zhang B, Tsiatis AA, Laber EB and Davidian M (2012). A robust method for estimating optimal treatment regimes. Biometrics 68 1010–1018. [PubMed: 22550953]

Zhang B, Tsiatis AA, Laber EB and Davidian M (2013). Robust estimation of optimal dynamic treatment regimes for sequential treatment decisions. Biometrika 100 681–694.

Zhang R, Dai B, Li L and Schuurmans D (2020). Gen{DICE}: Generalized Offline Estimation of Stationary Values. In International Conference on Learning Representations.

Zhao Y-Q, Zeng D, Laber EB and Kosorok MR (2015). New statistical learning methods for estimating optimal dynamic treatment regimes. Journal of the American Statistical Association 110 583–598. [PubMed: 26236062]

Zhao Y-Q, Laber EB, Ning Y, Saha S and Sands BE (2019). Efficient augmentation and relaxation learning for individualized treatment rules using observational data. Journal of Machine Learning Research 20 1–23.

Zhou X, Mayer-Hamblett N, Khan U and Kosorok MR (2017). Residual weighted learning for estimating individualized treatment rules. Journal of the American Statistical Association 112 169–187. [PubMed: 28943682]

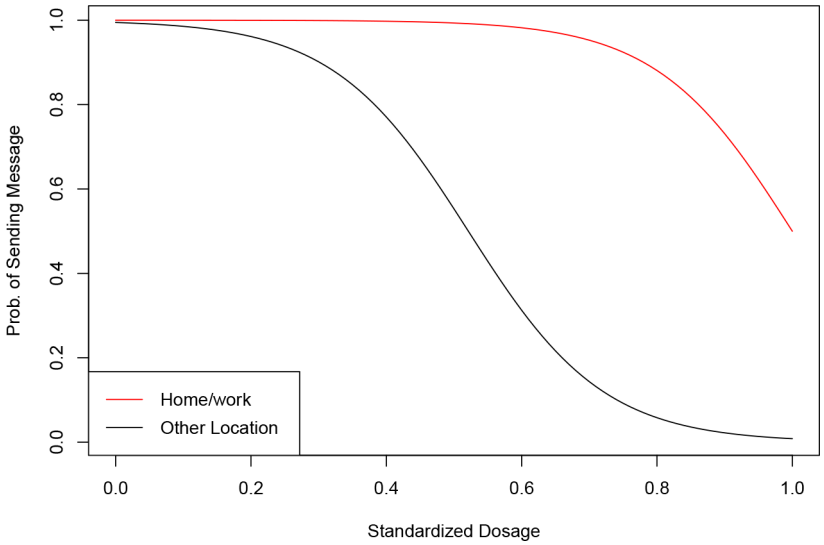

Fig 1.
The estimated policy in HeartSteps data.

Table 1

Monte Carlo estimation of the average rewards of the learned policy from proposed algorithm and three baseline offline RL algorithms over $T = 1000$ *with* 100 *replications. Numbers in parentheses are corresponding standard deviations. The oracle in-class optimal average rewards for both scenario are about 10.002.*

	n	***T***	**Our method**	**BEAR**	**BCQ**	**FQI**	**V-learning**
Scenario 1	40	50	9.215 (0.133)	7.513 (0.044)	8.187 (0.067)	9.728 (0.007)	9.246 (0.470)
	40	100	9.913 (0.050)	6.949 (0.053)	7.362 (0.068)	9.820 (0.028)	9.345 (0.476)
	80	50	9.834 (0.052)	7.487 (0.033)	7.992 (0.059)	9.764 (0.005)	9.461 (0.457)
Scenario 2	40	50	9.243 (0.133)	9.128 (0.009)	9.426 (0.020)	9.579 (0.028)	9.834 (0.097)
	40	100	9.905 (0.006)	9.508 (0.006)	9.692 (0.012)	9.858 (0.011)	9.840 (0.108)
	80	50	9.919 (0.005)	9.141 (0.011)	9.384 (0.020)	9.652 (0.025)	9.873 (0.097)