JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2021, VOL. 116, NO. 533, 392–409: Theory and Methods Special Issue on Precision Medicine and Individualized Policy Discovery
https://doi.org/10.1080/01621459.2020.1831925

Taylor & Francis
Taylor & Francis Group

Learning When-to-Treat Policies

Xinkun Nie[a], Emma Brunskill[a], and Stefan Wager[b]

[a]Department of Computer Science, Stanford University, Stanford, CA; [b]Graduate School of Business, Stanford University, Stanford, CA

ABSTRACT
Many applied decision-making problems have a dynamic component: The policymaker needs not only to choose whom to treat, but also when to start which treatment. For example, a medical doctor may choose between postponing treatment (watchful waiting) and prescribing one of several available treatments during the many visits from a patient. We develop an "advantage doubly robust" estimator for learning such dynamic treatment rules using observational data under the assumption of sequential ignorability. We prove welfare regret bounds that generalize results for doubly robust learning in the single-step setting, and show promising empirical performance in several different contexts. Our approach is practical for policy optimization, and does not need any structural (e.g., Markovian) assumptions. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received June 2019
Accepted September 2020

KEYWORDS
Causal inference; Minimax regret; Reinforcement learning; Sequential ignorability

1. Introduction

The promise of personalized data-driven decision-making has led to a surge in interest in methods that leverage observational data to help inform how and to whom interventions should be applied (Manski 2004; Zhang et al. 2012; Zhao et al. 2012; Dudík et al. 2014; Swaminathan and Joachims 2015; Elmachtoub and Grigas 2017; Kallus and Zhou 2018; Kitagawa and Tetenov 2018; Athey and Wager 2020; Bertsimas and Kallus 2020). Any solution to this policy learning problem needs to deal with numerous difficulties, including how to incorporate robustness to potential selection bias as well as fairness constraints articulated by stakeholders, and there have been several notable advances that address these difficulties over the past few years.

One limitation of this line of work, however, is that the results cited above all focus on a static setting where a decision-maker only sees each subject once and immediately decides how to treat the subject. In contrast, many problems of applied interest involve a dynamic component whereby the decision-maker makes a series of decisions based on time-varying covariates. In medicine, if a patient has a disease for which all known cures are invasive and have serious side effects, their doctor may choose to monitor disease progression for some time before prescribing one of these invasive treatments. As another example, a health inspector needs to not only choose which restaurants to inspect, but also when to carry out these inspections.

In this article, we study the problem of learning dynamic when-to-treat policies, where a decision-maker is only allowed to act once, but gets to choose both which action to take and when to perform the action.[1] This setting covers several application areas that have recently been discussed in the literature, including when to start antiretroviral therapy for HIV-positive patients to prevent AIDS while mitigating side effects (When To Start Consortium 2009), when to recommend mothers to stop breastfeeding to maximize infants' health (Moodie, Platt, and Kramer 2009), and when to to turn off ventilators for intensive care patients to maximize health outcomes (Prasad et al. 2017).

The available literature has developed several methods for evaluating and learning dynamic treatment rules from prior data, with notable contributions from statistics and epidemiology communities including from Murphy (2003), Robins (2004), Murphy (2005), Luckett et al. (2020), Tsiatis et al. (2019), Zhang et al. (2013, 2018), Van der Laan and Rose (2018, chap. 4), and the batch reinforcement learning community such as Jiang and Li (2016) and Thomas and Brunskill (2016). As discussed further below, these papers develop general approaches that can be used with arbitrary dynamic treatment rules. Here, in contrast, we seek to exploit special structure of the when-to-treat problem to develop tailored learning methods with desirable statistical and computational properties.

In developing our approach, we build on recent results on doubly robust (DR) static policy learning (Zhou, Athey, and Wager 2018; Athey and Wager 2020), and show how they can be adapted to our dynamic setting without making any structural (e.g., Markovian) assumptions and without compromising computational performance. Throughout this article, we assume sequential ignorability, meaning that any confounders that affect making a treatment choice at time t have already been measured by time t. Sequential ignorability is a widely used generalization of the classical

CONTACT Xinkun Nie ✉ *xinkun@stanford.edu* Department of Computer Science, Stanford University, Stanford, CA 94305.

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

[1] We note that the policies of interest in this article also include when-to-stop policies. By flipping the treatment indicator, it is without loss of generality that we only consider when-to-treat policies.

ignorability assumption of Rosenbaum and Rubin (1983) to the dynamic setting (Robins 1986, 2004; Hernán, Brumback, and Robins 2001; Murphy 2003). We develop methods that can leverage generic machine learning estimators of various nuisance components (e.g., the propensity of starting treatment in any given state and time) for learning policies with strong utilitarian regret bounds that hold in a nonparametric setting.

Our problem setting is closely related to batch reinforcement learning (Sutton and Barto 2018). The types of guarantees we derive, however, are more closely related to results from the static policy learning literature, in that we use tools from semiparametric statistics to derive sharp regret bounds given only nonparametric assumptions. To our knowledge, the reinforcement learning literature has not pursued nor obtained the type of results we achieve here for off-policy policy learning in a nonparametric setting.

We also note work on optimal stopping motivated by the problem of when to buy or sell an asset. This setting, however, is different from ours in that most of the literature on optimal stopping either works with a known probabilistic model (Van Moerbeke 1976; Jacka 1991), or assumes that we can observe the price evolution of the asset whether or not we purchase it (Goel, Dann, and Brunskill 2017). In contrast, we work in a nonparametric setting, and adopt a potential outcomes model in which we only get to observe outcomes corresponding to the sequence of actions we choose to take (Robins 1986; Imbens and Rubin 2015). We will review the related literature in more detail in Section 2.4 after first presenting our method below.

2. Policy Learning Under Sequential Ignorability

2.1. Setup and Notation

We work in the following statistical setting. We observe a set of $i = 1, \ldots, n$ independent and identically distributed trajectories generated from some distribution $\mathbb{P}$ that describe the evolution of subjects over T time steps. For each subject i, we observe a vector of states $S^{(i)} \in \mathcal{S}^T$ and actions $A^{(i)} \in \mathcal{A}^T$, as well as a final outcome $Y^{(i)} \in \mathbb{R}$.[2] For each $t = 1, \ldots, T$, $S_t^{(i)}$ denotes the state of the subject at time t and $A_t^{(i)}$ denotes the action taken. We write the set of possible actions as $\mathcal{A} = \{0, 1, \ldots, K\}$, and let $A_t = 0$ denote no action (i.e., no treatment assignment) at time t. We define the filtration $\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \cdots \subseteq \mathcal{F}_{T+1}$, where $\mathcal{F}_t = \sigma\left(S_{1:t}, A_{1:t-1}\right)$ contains information available at time t for $t = 1, \ldots, T$, and $\mathcal{F}_{T+1} = \sigma\left(S_{1:T}, A_{1:T}, Y\right)$ also has information on the final outcome. For notational convenience, we denote $S_{t_1:t_2}^{(i)} := \{S_{t_1}^{(i)}, \ldots, S_{t_2}^{(i)}\}$, and we similarly define $A_{t_1:t_2}^{(i)}$, and write the relevant generalization of the propensity score as $e_{t,a}(s_{1:t}) = \mathbb{P}\left[A_t = a \,\middle|\, S_{1:t} = s_{1:t}, A_{1:(t-1)} = 0\right]$.

We formulate causal effects in terms of potential outcomes (Neyman 1923; Rubin 1974; Robins 1986). For any set of actions $a \in \mathcal{A}^T$, we posit potential outcomes $Y^{(i)}(a_{1:T})$ and $S_t^{(i)}(a_{1:(t-1)})$ corresponding to the outcome and state values we would have obtained for subject i had we assigned treatment sequence a. To identify causal effects, we make the standard assumptions of sequential ignorability, consistency and overlap (Robins 1986, 2004; Hernán, Brumback, and Robins 2001; Murphy 2003).

Assumption 1 (Consistency of potential outcomes). Our observations are consistent with potential outcomes, in the sense that $Y^{(i)} = Y^{(i)}(A_{1:T}^{(i)})$ and $S_t^{(i)} = S_t^{(i)}(A_{1:(t-1)}^{(i)})$.

Assumption 2 (Sequential ignorability). Actions cannot respond to future information, that is, $\{Y(A_{1:(t-1)}, a_{t:T}), S_{t'}(A_{1:(t-1)}, a_{t:(t'-1)})\}_{t'=t+1}^T \perp\!\!\!\perp A_t \,\middle|\, \mathcal{F}_t$ for all $t = 1, \ldots, T$.

Assumption 3 (Overlap). There are constants $0 < \eta, \eta_0 < 1$ such that, for all $t = 1, \ldots, T$ and $s_{1:t} \in \mathcal{S}^t$, the following hold: $e_{t,a}(s_{1:t}) > \eta/T$ for all $a \in \mathcal{A} \setminus \{0\}$ and $e_{t,0}(s_{1:t}) > 1 - \eta_0/T$.

A policy π is a function that, for each time $t = 1, \ldots, T$, maps time-t observables to an action: $\pi_t : \mathcal{S}^t \times \mathcal{A}^{t-1} \rightarrow \mathcal{A}$ such that π_t is $\mathcal{F}_t$-measurable; then $\pi := \{\pi_t\}_{t=1}^T$. Recall that we focus on when-to-treat type rules, meaning that the decision-maker only gets to act once by starting a nonzero treatment regime at the time of their choice. For example, if $K = 3$ and $T = 5$, then the decision-maker may choose for instance to start treatment option #2 at time $t = 4$, resulting in a trajectory $A = (0, 0, 0, 2, 2)$.

When applying π on-policy, the behavior of π is fully characterized by the time at which π chooses to act, denoted by $\tau_\pi = \inf\{t : \pi_t(\cdot, \cdot) \neq 0\}$, and the treatment chosen, denoted by $W_\pi = \pi_{\tau_\pi}(\cdot, \cdot)$. When π chooses to never initiate treatment, we use the convention that $\tau_\pi = T + 1$ and $W_\pi = 0$. Note that τ_π and W_π are both $\mathcal{F}_{\tau_\pi}$-measurable.[3] For completeness, we also need to specify how π behaves off-policy, that is, how π would prescribe treatment along a trajectory whose past action or treatment sequence may disagree with π; and, in this article, we do so by assuming that π is regular in the sense of Definition 1.

Definition 1 (Regular policy). A regular when-to-treat policy π is determined by an $\mathcal{F}_t$-measurable stopping time τ_π and an associated $\mathcal{F}_{\tau_\pi}$-measurable decision variable $W_\pi \in \{1, \ldots, K\}$ as follows: For each time $t = 1, \ldots, T$, if $A_{t-1} \neq 0$ then $\pi_t(S_{1:t}, A_{1:(t-1)}) = A_{t-1}$, else if $t \geq \tau_\pi$ then $\pi_t(S_{1:t}, A_{1:(t-1)}) = W_\pi$, else $\pi_t(S_{1:t}, A_{1:(t-1)}) = 0$.

The main substance of Definition 1 is that we assume that if π suggests to start treatment k at a given moment, it persists in this choice k even if we fail to start treatment immediately. One notable limitation of this regularity condition is in the setting where some patients may die or otherwise be unable to receive treatment.[4] We discuss extensions of our approach beyond regular policies in Section 4.

Following Murphy (2005), we let $f_t(S_t \mid S_{1:(t-1)}, A_{1:(t-1)})$ be the conditional density for state transitions at time t. We can

[2] We note that it is without loss of generality that we assume the outcome $Y^{(i)}$ is only observed at the end of a trajectory, since intermediate outcomes/rewards can be incorporated as part of the state representation.

[3] See also Athey and Imbens (2018) for a closely related discussion of potential outcomes in the context of staggered adoption.

[4] For example, consider a case where π says we should have started treatment on day 5 but we did not in fact start treatment then, that is, $A_5 = 0$, and then the patient dies on day 6. Here, realistically, π should recognize that starting treatment is now impossible and prescribe $\pi_6(S_{1:6}, A_{1:5}) = 0$; however, doing so would be inconsistent with Definition 1.

write the distribution function for trajectories $(s_{1:T}, a_{1:T})$ as

$$f(s,\, a) = f(s_1)\, \mathbb{P}\left[A_1 \,\middle|\, s_1\right] \prod_{t=2}^{T} f_t\left(s_t \,\middle|\, s_{1:(t-1)}, a_{1:(t-1)}\right)$$
$$\mathbb{P}\left[A_t = a_t \,\middle|\, s_{1:t}, a_{1:(t-1)}\right], \tag{1}$$

and denote the expectation with respect to this distribution as $\mathbb{E}$. Moreover, by Assumption 2, the distribution of a trajectory under policy π is

$$f(s,\, a;\, \pi) = f(s_1) \mathbb{1}_{a_1 = \pi_1(s_1)} \prod_{t=2}^{T} f_t\left(s_t \,\middle|\, s_{1:(t-1)}, a_{1:(t-1)}\right)$$
$$\mathbb{1}_{a_t = \pi_t(s_{1:t}, a_{1:(t-1)})}, \tag{2}$$

and we use $\mathbb{E}_\pi$ to denote the expectation with respect to it. Define

$$V_\pi := \mathbb{E}_\pi\left[Y\right] = \mathbb{E}\left[Y\left(\pi_1(S_1), \pi_2\left(S_1,\, S_2(\pi_1(S_1)), \pi_1(S_1)\right), \ldots\right)\right] \tag{3}$$

to be the value of the policy π, that is, the expected outcome Y with actions A_t chosen according to π such that $A_t = \pi_t(S_{1:t},\, A_{1:(t-1)})$ for all $t = 1, \ldots, T$. We further define the conditional value function

$$\begin{aligned}&\mu_{\pi,t}(s_{1:t}, a_{1:t-1})\\ &= \mathbb{E}_\pi\left[Y \,\middle|\, S_{1:t} = s_{1:t}, A_{1:t-1} = a_{1:t-1}\right]\\ &= \mathbb{E}\Bigg[Y(a_{1:t-1}, \pi_t(S_{1:t}, A_{1:t-1}), \pi_{t+1}(\ldots), \ldots) \,\Big|S_{1:t}\\ &\qquad = s_{1:t}, A_{1:t-1} = a_{1:t-1}\Bigg],\end{aligned} \tag{4}$$

and the Q-function

$$\begin{aligned}&Q_{\pi,t}(s_{1:t}, a_{1:t})\\ &= \mathbb{E}_\pi\left[Y \,\middle|\, S_{1:t} = s_{1:t}, A_{1:t} = a_{1:t}\right]\\ &= \mathbb{E}\Bigg[Y(a_{1:t}, \pi_{t+1}(S_{1:t}, S_{t+1}(A_{1:t}), A_{1:t}), \ldots) \,\Big|\\ &\qquad S_{1:t} = s_{1:t}, A_{1:t} = a_{1:t}\Bigg].\end{aligned} \tag{5}$$

For any class Π, we write the optimal value function as $V^* = \sup_{\pi \in \Pi} V_\pi$, and define the regret of a policy $\pi \in \Pi$ as $R(\pi) = V^* - V_\pi$ (Manski 2004). Given this setting, our goal is to learn the best policy from a predefined policy class Π to minimize regret. Our main result is a method for learning a policy $\hat{\pi} \in \Pi$ along with a bound on its regret $R(\hat{\pi})$.

2.2. Existing Methods

In the static setting, a popular approach to policy learning starts by first providing an estimator $\widehat{V}_\pi$ for the value V_π of each feasible policy $\pi \in \Pi$, and then sets $\hat{\pi} = \operatorname{argmax}\{\widehat{V}_\pi : \pi \in \Pi\}$ (e.g., Manski 2004; Zhao et al. 2012; Swaminathan and Joachims 2015; Kitagawa and Tetenov 2018; Athey and Wager 2020). At a high level our goal is to pursue the same strategy, but now in a dynamic setting. The challenge is then to find a robust estimator $\widehat{V}_\pi$ that behaves well when optimized over a policy class Π of interest—both statistically and computationally.

Perhaps the most straightforward approach to estimating V_π starts from inverse propensity weighting as used in the context of marginal structural modeling (Precup 2000; Robins, Hernán, and Brumback 2000). Given sequential ignorability, we can write inverse propensity weights $\gamma_t^{(i)}(\pi)$ for any policy π recursively as follows, resulting in a value estimate

$$\hat{V}_\pi^{\mathrm{IPW}} = \frac{1}{n} \sum_{i=1}^{n} \gamma_T^{(i)}(\pi) Y^{(i)},$$
$$\gamma_t^{(i)}(\pi) = \frac{\gamma_{t-1}^{(i)} \mathbb{1}\left(\left\{A_t^{(i)} = \pi(S_{1:t}^{(i)}, A_{1:t-1}^{(i)})\right\}\right)}{\mathbb{P}\left[A_t^{(i)} = \pi(S_{1:t}^{(i)}, A_{1:t-1}^{(i)}) \,\middle|\, S_{1:t}^{(i)}, A_{1:t-1}^{(i)}\right]}, \tag{6}$$

or the normalized alternative $\hat{V}_\pi^{\mathrm{WIPW}} = \sum_{i=1}^n \gamma_T^{(i)}(\pi) Y^{(i)} \big/ \sum_{i=1}^n \gamma_T^{(i)}(\pi)$. The functional form of $\hat{V}_\pi^{\mathrm{IPW}}$ makes it feasible to optimize this value estimate over a prespecified policy class $\pi \in \Pi$ (e.g., via a grid-search or mixed integer programming). By Assumptions 1–3, IPW is consistent if the treatment probabilities are known a priori, and by uniform concentration arguments following Kitagawa and Tetenov (2018), the regret of the policy $\hat{\pi}$ learned by maximizing $\hat{V}_\pi^{\mathrm{IPW}}$ over $\pi \in \Pi$ decays as $1/\sqrt{n}$ if Π is not too large (e.g., if Π is a VC-class). Zhao et al. (2015) provided regret bounds for nonparametric IPW-type estimators in a dynamic setting.

While inverse propensity weighting is a simple and transparent approach to estimating V_π, it has several limitations. In observational studies treatment probabilities need to be estimated from data, and it is known that the variant of (6) with estimated weights $\hat{\gamma}_t^{(i)}(\pi)$ can perform poorly with even mild estimation error (see, e.g., Y. Liu et al. 2018). Furthermore, for any policy π considered, the IPW value estimator only uses trajectories that match the policy π exactly, which can make policy learning sample-inefficient. Finally, the IPW estimator is known to be unstable when treatment propensities are small, and this difficulty is exacerbated in the multi-period setting as the probability of observing any specific trajectory decays, which can cause challenges during policy learning (Doroudi, Thomas, and Brunskill 2017). In the static (i.e., single time step) policy learning setting, related considerations led several authors to recommend against inverse propensity weighted policy learning and to develop new methods that were found to have stronger properties both in theory and in practice (Zhang et al. 2012; Dudík et al. 2014; Zhou et al. 2017; Kallus 2018; Athey and Wager 2020).

Another approach to estimating V_π is using a DR estimator as follows (Zhang et al. 2013; Jiang and Li 2016; Thomas and Brunskill 2016)

$$\widehat{V}_\pi^{\mathrm{DR}} = \frac{1}{n} \sum_{i=1}^{n} \Bigg(\hat{\gamma}_T^{(i)}(\pi) Y^{(i)} - \sum_{t=1}^{T} \left(\hat{\gamma}_t^{(i)}(\pi) - \hat{\gamma}_{t-1}^{(i)}(\pi)\right)$$
$$\hat{\mu}_\pi\left(S_{1:t}^{(i)}, A_{1:t-1}^{(i)}\right)\Bigg), \tag{7}$$

where $\hat{\mu}_\pi(\cdot)$ is an estimator of $\mu_\pi(\cdot)$, the expected value following policy π conditionally on the history up to time t as defined in the previous subsection. This estimator generalizes the well known augmented inverse propensity weighted estimator of Robins, Rotnitzky, and Zhao (1994) beyond the static case. The DR estimator (7) is consistent if either the propensity weights $\{\hat{\gamma}_t(\cdot)\}_{t=1}^T$ or the conditional value estimates $\hat{\mu}_\pi(\cdot)$ are consistent. For a further discussion of DR estimation under sequential ignorability, see Van der Laan and Rose (2018, chap. 4), Tsiatis et al. (2019), and references therein.

From an optimization point of view, a limitation of (7) is that evaluating a given policy π requires nuisance components estimates $\hat{\mu}_\pi(\cdot)$ that are specific to the policy under consideration. This makes policy learning by optimizing $\widehat{V}_\pi^{\text{DR}}$ challenging for several reasons. Computationally, maximizing $\widehat{V}_\pi^{\text{DR}}$ for all π in a nontrivial set Π would require solving a multitude of nonparametric dynamic programming problems.[5] Statistically, standard regret bounds for policy learning for single time step problems rely crucially on the fact that $\widehat{V}_\pi$ is continuous in π in an appropriate sense, meaning that two policies are taken to have similar values if they make similar recommendations in almost all cases (see, e.g., Athey and Wager 2020). But, if $\hat{\mu}_\pi(\cdot)$ is learned separately for each π, we have no reason to believe that two similar policies would necessarily have similar value function estimates—unless one were to use specially designed $\hat{\mu}_\pi(\cdot)$ estimators.[6]

An alternate popular approach is to perform policy learning by estimating policy values using Q-learning and then finding the action or treatment decision that maximizes the resulting value estimate. Such methods can suffer from model misspecification, and from error amplification due to the recursive structure of the algorithm. Our contribution focuses on alternate more robust methods, and we discuss Q-learning approaches in the related work section.

2.3. Advantage DR Policy Learning

The goal of this article is to develop a new method for policy learning that addresses the shortcomings of both inverse propensity weighting and the doubly robust method discussed above in the case of when-to-treat policies.[7] Our main proposal, the advantage doubly robust (ADR) estimator, uses an outcome regression like the DR estimator (7) to stabilize and robustify its value estimates. However, unlike the estimator (7) which needs to use different outcome regressions $\hat{\mu}_\pi(\cdot)$ to evaluate each different policy π, ADR only has "universal" nuisance components that do not depend on the policy being estimated, leveraging the when-to-treat (or when-to-stop) structure of the domain. Throughout this article, we will find that this universality property enables us to both effectively optimize our value estimates to learn policies and to prove robust utilitarian regret bounds.

The motivation for our approach starts from an "advantage decomposition" presented below. First, define

$$\begin{aligned}\mu_{\text{now},k}(s_{1:t},\, t) &:= \mathbb{E}\left[Y \,\middle|\, S_{1:t} = s_{1:t}, A_{1:t-1} = 0, A_t = k\right],\\ \mu_{\text{next},k}(s_{1:t},\, t) &:= \mathbb{E}\Big[\mu_{\text{now},k}\left(S_{1:t+1},\, t+1\right) \,\Big|\, S_{1:t} = s_{1:t},\\ &\qquad A_{1:t} = 0\Big],\end{aligned} \tag{8}$$

which measure the conditional value of a policy that starts treatment k either now or in the next time period, given that we have not yet started any treatment. Note that, for any when-to-treat policy π as considered in this article, the expectations in (8) do not depend on π because the conditioning specifies all actions from time $t = 1$ to T. Given policies $\pi, \pi' \in \Pi$, define $\Delta(\pi, \pi') = V_\pi - V_{\pi'}$ to be the difference in value of the two policies. Denote the never-treating policy by $\mathbf{0}$. Then, a result from Kakade (2003, chap. 5) and Murphy (2005) yields the following.

Lemma 1. Under Assumptions 1 and 2 let π be a regular when-to-treat policy in the sense of Definition 1. Then

$$\Delta(\pi, \mathbf{0}) = \mathbb{E}_{\mathbf{0}}\left[\sum_{t=\tau_\pi}^{T} \mu_{\text{now},W_\pi}(S_{1:t}, t) - \mu_{\text{next},W_\pi}(S_{1:t}, t)\right], \tag{9}$$

where $\mathbb{E}_{\mathbf{0}}$ samples trajectories under a never-treating policy and, following Definition 1, τ_π is the time at which π starts treating and W_π is the treatment chosen at that time.

Proof. Given our setup, Lemma 1 of Murphy (2005) implies that

$$\begin{aligned}\Delta(\pi, \mathbf{0}) &= -\mathbb{E}_{\mathbf{0}}\left[\sum_{t=1}^{T} Q_{\pi,t}(S_{1:t}, A_{1:t}) - \mu_{\pi,t}(S_{1:t}, A_{1:t-1})\right]\\ &= -\mathbb{E}_{\mathbf{0}}\Bigg[\sum_{t=1}^{T} \mathbb{1}_{t \geq \tau_\pi}\Big(Q_{\pi,t}(S_{1:t}, \mathbf{0}_{1:t})\\ &\qquad - \mu_{\pi,t}(S_{1:t}, \mathbf{0}_{1:(t-1)})\Big)\Bigg].\end{aligned} \tag{10}$$

Because π is a regular when-to-stop policy, whenever $t \geq \tau_\pi$, the policy π prescribes starting treatment W_π immediately if no other treatment has been started yet, that is,

$$\begin{aligned}\mathbb{1}_{t \geq \tau_\pi}\, \mu_{\pi,t}(S_{1:t}, \mathbf{0}_{1:(t-1)}) &\overset{(a)}{=} \mathbb{1}_{t \geq \tau_\pi} \mathbb{E}\Big[Y(\mathbf{0}_{1:(t-1)}, W_\pi, W_\pi, \ldots) \,\Big|\\ &\qquad S_{1:t}, A_{1:t-1} = \mathbf{0}_{1:(t-1)}\Big]\\ &\overset{(b)}{=} \mathbb{1}_{t \geq \tau_\pi} \mathbb{E}\Big[Y(\mathbf{0}_{1:(t-1)}, W_\pi, W_\pi, \ldots) \,\Big|\\ &\qquad S_{1:t}, A_{1:t-1} = \mathbf{0}_{1:(t-1)}, A_t = W_\pi\Big]\\ &\overset{(c)}{=} \mathbb{1}_{t \geq \tau_\pi} \mathbb{E}\Big[Y \,\Big|\, S_{1:t}, A_{1:t-1}\\ &\qquad = \mathbf{0}_{1:(t-1)}, A_t = W_\pi\Big]\\ &= \mathbb{1}_{t \geq \tau_\pi}\, \mu_{\text{now},W_\pi}(S_{1:t}, t),\end{aligned}$$

[5] See Zhang et al. (2015) for an example of using decision lists for policy optimization with a doubly robust estimator.

[6] One heuristic solution to this difficulty, proposed by Zhang et al. (2013), is to first derive a policy estimate $\hat{\pi}^*$ via, for example, IPW or fitted-Q learning, and then to use value estimates $\hat{\mu}_{\hat{\pi}^*}(\cdot)$ to evaluate all policies $\pi \in \Pi$ using (7). The advantage of this proposal is that learning by maximizing $\widehat{V}_\pi^{\text{DR}}$ becomes more tractable, since one does not need to refit nuisance components to evaluate different policies.

[7] We emphasize that the IPW and DR estimators discussed above can be used with general dynamic policies; in contrast, our method can only be used for learning when-to-treat policies. Our proposed method does not present an alternative to IPW or DR in the general case.

where (a), (b), and (c) are by Definition 1, Assumption 2, and Assumption 1, respectively. Furthermore, given our definition of regular policies, we know that if $t \geq \tau_\pi$ and $A_t = 0$, then π will deterministically prescribe treatment W_π at time $t+1$ regardless of the state S_{t+1}, and so

$$\begin{aligned}
&\mathbb{1}_{t\geq\tau_\pi}\, Q_{\pi,t}(S_{1:t},\, \mathbf{0}_{1:t})\\
&\quad= \mathbb{1}_{t\geq\tau_\pi} \int \mathbb{E}_\pi\left[Y \,\middle|\, S_{1:t+1},\, A_{1:t} = \mathbf{0}_{1:t}\right]\\
&\qquad dF_{t+1}\left(S_{t+1} \,\middle|\, S_{1:t},\, A_{1:t} = \mathbf{0}_{1:t}\right)\\
&\quad\overset{(d)}{=} \mathbb{1}_{t\geq\tau_\pi} \int \mathbb{E}\left[Y(\mathbf{0}_{1:t}, W_\pi, W_\pi, \ldots) \,\middle|\, S_{1:t+1},\, A_{1:t} = \mathbf{0}_{1:t}\right]\\
&\qquad dF_{t+1}\left(S_{t+1} \,\middle|\, S_{1:t},\, A_{1:t} = \mathbf{0}_{1:t}\right)\\
&\quad\overset{(e)}{=} \mathbb{1}_{t\geq\tau_\pi} \int \mathbb{E}\Big[Y(\mathbf{0}_{1:t}, W_\pi, W_\pi, \ldots) \,\Big|\, S_{1:t+1},\\
&\qquad A_{1:t} = \mathbf{0}_{1:t}, A_{t+1} = W_\pi\Big]\\
&\qquad\quad \times dF_{t+1}\left(S_{t+1} \,\middle|\, S_{1:t},\, A_{1:t} = \mathbf{0}_{1:t}\right)\\
&\quad\overset{(f)}{=} \mathbb{1}_{t\geq\tau_\pi} \int \mathbb{E}\left[Y \,\middle|\, S_{1:t+1},\, A_{1:t} = \mathbf{0}_{1:t}, A_{t+1} = W_\pi\right]\\
&\qquad dF_{t+1}\left(S_{t+1} \,\middle|\, S_{1:t},\, A_{1:t} = \mathbf{0}_{1:t}\right),
\end{aligned}\tag{11}$$

where (d), (e), and (f) are by Definition 1, Assumption 2 and Assumption 1, respectively. The conclusion (9) emerges by plugging these facts into (10). □

In Lemma 1, the expectation is taken with respect to the never-treating policy $\mathbf{0}$. To make this result usable in practice, the following lemma translates it in terms of expectations taken with respect to the sampling measure. Recall that the propensity of starting treatment a assuming a never-treating history up to time t is denoted by $e_{t,a}(s_{1:t}) = \mathbb{P}\left[A_t = a \,\middle|\, S_{1:t} = s_{1:t}, A_{1:t-1} = 0\right]$. The proof of Lemma 2, given in Appendix A in the supplementary materials, follows directly from a change of measure.

Lemma 2. In the setting of Lemma 1,

$$\begin{aligned}
\Delta(\pi, \mathbf{0}) = \mathbb{E}\Bigg[\sum_{t=1}^{T} \mathbb{1}_{t\geq\tau_\pi} \frac{\mathbb{1}_{A_{1:t-1}=0}}{\prod_{t'=1}^{t-1} e_{t',0}(S_{1:t'})}\\
\left(\mu_{\text{now},W_\pi}(S_{1:t}, t) - \mu_{\text{next},W_\pi}(S_{1:t}, t)\right)\Bigg].
\end{aligned}\tag{12}$$

This representation (12) is at the core of our approach, as it decomposes the relative value of any given policy π in comparison to that of the never-treating policy $\mathbf{0}$ into a sum of local advantages. For any t, the local advantage

$$\delta_{\text{local},k}(s_{1:t}, t) := \mu_{\text{now},k}(s_{1:t}, t) - \mu_{\text{next},k}(s_{1:t}, t) \tag{13}$$

is the relative advantage of starting treatment k at t versus at $t+1$ given the state history $s_{1:t}$. The upshot is that the specification of these local advantages does not depend on which policy we are evaluating, so if we get a handle on quantities $\delta_{\text{local},k}(s_{1:t}, t)$ for all s and t, we can use (12) to evaluate any policy π.

Note that the quantity defined in (13) can be seen as a specific treatment effect, namely the effect of starting treatment k at time t versus $t+1$ among all trajectories that were in state $s_{1:t}$ at time t and started treatment k in either time t or $t+1$. Given this observation, we propose turning (12) into a feasible estimator by replacing all instances of the unknown regression surfaces $\delta_{\text{local},k}(s_{1:t}, t)$ with DR scores analogous to those used for augmented inverse propensity weighting in the static case (Robins and Rotnitzky 1995).

```
Algorithm 1: Advantage doubly robust (ADR) estimator
1 Estimate the outcome models $\mu_{\text{now},k}(\cdot)$, $\mu_{\text{next},k}(\cdot)$, as well
  as treatment propensities $e_{t,a}(s_{1:t})$ with cross-fitting using
  any supervised learning method tuned for prediction
  accuracy.
2 Given these nuisance component estimates, we construct
  value estimates
    $\hat{\Delta}(\pi, \mathbf{0}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{T} \mathbb{1}_{t\geq\tau_\pi^{(i)}} \frac{\mathbb{1}_{A^{(i)}_{1:t-1}=0}}{\prod_{t'=1}^{t-1} \hat{e}^{-q(i)}_{t',0}(S^{(i)}_{1:t'})} \hat{\Psi}_{t,W_\pi}(S^{(i)}_{1:t})$    (14)
  for each policy $\pi \in \Pi$, where the relevant DR score is
    $\hat{\Psi}_{t,k}(S^{(i)}_{1:t}) = \hat{\mu}^{-q(i)}_{\text{now},k}(S^{(i)}_{1:t}, t) - \hat{\mu}^{-q(i)}_{\text{next},k}(S^{(i)}_{1:t}, t)$
        $+ \mathbb{1}_{A^{(i)}_t = k} \frac{Y^{(i)} - \hat{\mu}^{-q(i)}_{\text{now},k}(S^{(i)}_{1:t}, t)}{\hat{e}^{-q(i)}_{t,k}(S^{(i)}_{1:t})}$
        $- \mathbb{1}_{A^{(i)}_t = 0}\mathbb{1}_{A^{(i)}_{t+1} = k} \frac{Y^{(i)} - \hat{\mu}^{-q(i)}_{\text{next},k}(S^{(i)}_{1:t}, t)}{\hat{e}^{-q(i)}_{t,0}(S^{(i)}_{1:t})\hat{e}^{-q(i)}_{t+1,k}(S^{(i)}_{1:t+1})}.$    (15)
3 Learn the optimal policy by setting
  $\hat{\pi} = \text{argmax}_{\pi\in\Pi}\hat{\Delta}(\pi, \mathbf{0})$.
```

More specifically, we propose the following 3-step policy learning algorithm, outlined as Algorithm 1. We call our approach the ADR estimator, because it replaces local advantages (13) with appropriate DR scores (15) when estimating $\Delta(\pi, \mathbf{0})$. In the first estimation step in Algorithm 1, we employ cross-fitting where we divide the data into Q folds, and only use the $Q-1$ folds that a sample trajectory does not belong to learn the estimates of its nuisance components; we use superscript $-q(i)$ on a predictor to denote using trajectories of all folds excluding the fold that the ith trajectory belongs to in training a predictor.[8]

The main strength of this procedure relative to existing DR approaches discussed above (Zhang et al. 2013; Jiang and Li 2016; Thomas and Brunskill 2016) is that ADR can evaluate any stopping policy using universal scores $\hat{\Psi}_{t,k}(\cdot)$ that do not depend on π. This allows us to ensure smoothness criteria we use to provide regret bounds for policy optimization. It also provides computational benefits for using the ADR estimator for policy optimization: the specific policy π we are evaluating only enters into (14) by specifying which DR scores we should sum over.

[8] The idea of cross-fitting has gained growing popularity recently to reduce the effect of own-observation bias and to enable results on semiparametric rates of convergence using generic nuisance component estimates (Schick 1986; Chernozhukov et al. 2018; Athey and Wager 2020).

In particular, the number of nuisance components we need to learn in the first step of the ADR procedure scales linearly with the horizon T, but not with the complexity of the policy class Π.

By constructing DR scores $\hat{\Psi}_{t,k}(\cdot)$, the ADR estimator benefits from certain robustness properties; however, it is not DR in the usual sense, for example, we do not robustly correct for the change of measure used to get from the representation in Lemma 1 to the one in Lemma 2. We discuss the asymptotic behavior of our method in Section 3.

In our experiments, we learn all the nuisance components in the first step with nonparametric regression methods (e.g., boosting, lasso, a deep net, etc.), and then optimize for the best in-class policy by performing a grid search over the parameters that define the policies in a policy class of interest.

Remark 3. For the purpose of estimating $\mu_{\text{next},k}(s_{1:t},t)$, it is helpful to re-express it in terms of conditional expectations. First, under Assumption 3, we can continue from (11) and rewrite $\mu_{\text{next},k}(s_{1:t},t)$ via inverse-propensity weighting as

$$\mu_{\text{next},k}(s_{1:t},t) = \mathbb{E}\left[\frac{\mathbb{1}_{A_{t+1}=k}}{e_{t+1,k}(S_{1:t+1})} Y \,\middle|\, S_{1:t},\, A_{1:t} = \mathbf{0}_{1:t}\right]. \quad (16)$$

Then, using Bayes' rule, we can verify that

$$\mu_{\text{next},k}(s_{1:t},t) = \frac{\mathbb{E}\left[Y/e_{t+1,k}(S_{1:t+1}) \,\middle|\, S_{1:t} = s_{1:t}, A_{1:t} = 0,\, A_{t+1} = k\right]}{\mathbb{E}\left[1/e_{t+1,k}(S_{1:t+1}) \,\middle|\, S_{1:t} = s_{1:t}, A_{1:t} = 0,\, A_{t+1} = k\right]}. \quad (17)$$

This last expression implies that we can consistently estimate $\mu_{\text{next},k}(\cdot,t)$ via weighted nonparametric regression of Y on $S_{1:t}$ on the set of observations with $A_{1:t} = 0$ and $A_{t+1} = k$, with weights $e^{-1}_{t+1,k}(S_{1:t+1})$. In practice, this may yield more stable estimates of $\mu_{\text{next},k}(s_{1:t},t)$ than an unweighted nonparametric regression with response $\mathbb{1}_{A_{t+1}=k}/e_{t+1,k}(S_{1:t+1})\, Y$.

2.4. Related Work

The problem of learning optimal dynamic sequential decision rules is also called learning optimal dynamic regimes (Murphy 2003; Robins 2004), adaptive strategies (Lavori and Dawson 2000), or batch off-policy policy learning in the reinforcement learning (RL) literature (Sutton and Barto 2018). As this is a large literature spanning several fields, a comprehensive overview of the literature on dynamic decision rules is beyond the scope of this article; instead, we refer the reader to recent review papers (Vansteelandt and Joffe 2014; Kosorok and Laber 2019; Clifton and Laber 2020; Levine et al. 2020), as well as textbooks by Chakraborty and Moodie (2013) and Tsiatis et al. (2019).

Our approach builds most directly on methods for structured policy learning in the static setting (Manski 2004; Zhao et al. 2012; Swaminathan and Joachims 2015; Kitagawa and Tetenov 2018; Athey and Wager 2020). From this perspective, the work of Zhao et al. (2015), who extend the outcome-weighted learning approach of Zhao et al. (2012) to the dynamic setting, is close to us in its statistical setting. Inverse-propensity type methods, including outcome-weighted learning and importance sampling, are transparent and simple to implement; however, they have sometimes been observed exhibit problematically high variance, especially in dynamic settings (see, e.g., Doroudi, Thomas, and Brunskill 2017). Here, we seek to improve on simple inverse-propensity weighting type methods, and to develop dynamic policy learning methods that can leverage outcome models for improved power. In doing so, we note that here are by now well established methods for DR dynamic policy evaluation under sequential ignorability (Zhang et al. 2013; Jiang and Li 2016; Thomas and Brunskill 2016), but unlike in the static setting it is not immediately clear how best to adapt these methods to a learning setting (see Athey and Wager 2020 for a general discussion of how to move from static policy evaluation to learning).

One different but very influential line of work on dynamic decision rules uses Q-learning (Robins 1986; Watkins and Dayan 1992; Ernst, Geurts, and Wehenkel 2005; Murphy 2005) which, at a high level, seeks to solve a noisy dynamic programming problem to learn an optimal policy (see Clifton and Laber 2020; Levine et al. 2020 for recent reviews). The biggest algorithmic difference between our setting and Q-learning is that, here, we seek to find the best policy within a constrained class, while the dynamic programming formulation of Q-learning is tailored to learning the best Q function in a constrained class, and then extracting a policy by taking the decision that maximizes the learned function. Policy search and our ADR estimator can be advantageous when there are predefined structural constraints on the policy class (e.g., for ease of interpretability, budget constraints, etc.). In the reinforcement learning community, actor-critic methods are popular which combine both a constrained policy class and a constrained Q function, but there has been limited theoretical analysis of their properties yet. It is also possible to use Q-learning for policy evaluation, and then to learn policies by taking an argmax of the value estimate over a policy class. For example, Zhang et al. (2018) utilized this strategy to learn decision rules that can be expressed as a sequence of simple "if-then" rules; see also Le, Voloshin, and Yue (2019). This approach, however, may be computationally demanding, as it involves separately estimating policy values via Q-learning for each candidate policy; in contrast, our approach can share computation across policy evaluations.

There is a long history of theoretical work on batch offline Q-learning in the reinforcement learning community (e.g., Munos 2003; Munos and Szepesvári 2008; Farahmand et al. 2016) focused on bounding the error of the returned policy during batch learning relative to the optimal policy. However, almost all such prior work assumes the domain is Markov, and requires that the optimal value function is realizable by the regressor function used to model the value function (realizability) and that the regressor function class is closed under the Bellman backup operator used during dynamic programming (completeness); see recent work by Chen and Jiang (2019) and Le, Voloshin, and Yue (2019).[9] Closer to our policy search work is recent theoretical work on batch policy search methods; however, such work has focused on settings with Markov structure. Liu et al. (2019) provided convergence guarantees for batch

[9] Such results also require a bound on the concentratability coefficient (Munos 2003), which measures the ratio of the state action distribution of a policy to the state-action distribution under the behavior policy, for any behavior policy Chen and Jiang (e.g., 2019); Le, Voloshin, and Yue (e.g., 2019); Munos and Szepesvári (e.g., 2008)—this can be viewed as a similar analogue to the overlap assumption as in Assumption 3.

policy gradient, and Kallus and Uehara (2020) provides regret bounds in the restricted setting where the policy value is a concave function of the policy parameters. When the Markov, realizability or completeness assumptions fail, past theoretical results tend to provide no bounds or an additional constant regret. Our work is motivated by healthcare and settings where such assumptions may fail, and we wish to compete with the best available solution in a given policy class.

Throughout this article, we assume that the time horizon T over which we can act remains bounded. In the reinforcement learning literature, this is typically referred to as a "finite horizon" setting, and there is also a large literature on the "infinite horizon" setting where T is not bounded (see, e.g., Antos, Szepesvári, and Munos 2008a, 2008b; Munos and Szepesvári 2008; Q. Liu et al. 2018; Uehara and Jiang 2019; Luckett et al. 2020). These cases are considerably different from ours and are beyond the scope of this work.

Finally, in the optimal stopping literature (e.g., Van Moerbeke 1976; Jacka 1991; Mordecki 2002; Goel, Dann, and Brunskill 2017), the treatment choices are binary, and the goal is to optimize for a policy for when to start or stop a treatment. Most work on optimal stopping assumes generator is available for the system dynamics, or the full potential outcomes are available in the training data. In contrast, in our setup, we assume that we can only observe rewards corresponding to actions taken in the training data. Rust (1987) considered the descriptive problem of fitting an optimal stopping model to the behavior of a rational agent; this is different from the problem of learning a decision rule that can be used to guide future decisions in this article.

3. Asymptotics

In this section, we study large-sample behavior of the ADR estimator proposed in Section 2.3 for policy learning in when-to-treat settings over a class of policies Π. It is now standard in the literature in static policy learning for policies over a single decision to bound regret over the learned policy (e.g., Manski 2004; Swaminathan and Joachims 2015; Kitagawa and Tetenov 2018; Athey and Wager 2020). However, to our knowledge there are no directly comparable results for the sequential decision process setting.

Following the literature on static policy learning, our main goal is to prove a bound on the utilitarian regret R of the learned policy $\hat{\pi}$, where

$$R\left(\hat{\pi}\right) = \sup\left\{V(\pi) : \pi \in \Pi\right\} - V\left(\hat{\pi}\right). \tag{18}$$

To do so, we follow the high-level proof strategy taken by Athey and Wager (2020) for studying static DR policy learning. We first consider the behavior of an "oracle" learner who runs our procedure but with perfect estimates of the nuisance components $\mu_{\text{now},k}(\cdot)$, $\mu_{\text{next},k}(\cdot)$, and $e_{t,k}(\cdot)$, then we couple the behavior of our feasible estimator that uses estimated nuisance components with this oracle.

Following this outline, recall that our approach starts by estimating the policy value difference $\Delta(\pi, \mathbf{0})$ between deploying policy π and the never treating policy $\mathbf{0}$. The oracle variant of our estimator $\widehat{\Delta}(\pi, \mathbf{0})$ is then

$$\tilde{\Delta}(\pi, \mathbf{0}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{T} \mathbb{1}_{t \geq \tau_\pi} \frac{\mathbb{1}_{A^{(i)}_{1:t-1}=0}}{\prod_{t'=1}^{t-1} e_{t',0}(S_{1:t'})} \tilde{\Psi}_{t,W_\pi}(S^{(i)}_{1:t}), \tag{19}$$

where

$$\begin{aligned}\tilde{\Psi}_{t,k}(S^{(i)}_{1:t}) = {} & \mu_{\text{now},k}(S^{(i)}_{1:t}, t) - \mu_{\text{next},k}(S^{(i)}_{1:t}, t) \\ & + \mathbb{1}_{A^{(i)}_t=k} \frac{Y^{(i)} - \mu_{\text{now},k}(S^{(i)}_{1:t}, t)}{e_{t,k}(S^{(i)}_{1:t})} \\ & - \mathbb{1}_{A^{(i)}_t=0} \mathbb{1}_{A^{(i)}_{t+1}=k} \frac{Y^{(i)} - \mu_{\text{next},k}(S^{(i)}_{1:t}, t)}{e_{t,0}(S^{(i)}_{1:t}) e_{t+1,k}(S^{(i)}_{1:t+1})}.\end{aligned} \tag{20}$$

We name (19) the *oracle* estimator since we assume $\mu_{\text{now},k}$, $\mu_{\text{next},k}$, $e_{t,k}$ for $t = 1, \ldots, T$ and $k = 1, \ldots, K$ take ground-truth values in (20).

Because the nuisance components in (19) are known a priori, we can use a standard central limit theorem argument to verify the following.

Lemma 4. Suppose that Assumptions 1–3 hold and that $|Y| \leq M$ almost surely for some constant M, for a fixed policy $\pi \in \Pi$,

$$\sqrt{n}(\tilde{\Delta}(\pi, \mathbf{0}) - \Delta(\pi, \mathbf{0})) \Rightarrow \mathcal{N}(0, \Omega_\pi),$$

$$\text{where} \quad \Omega_\pi = \text{var}\left[\sum_{t=1}^{T} \mathbb{1}_{t \geq \tau^{(i)}_\pi} \frac{\mathbb{1}_{A^{(i)}_{1:t-1}=0}}{\prod_{t'=1}^{t-1} e_{t',0}(S^{(i)}_{1:t'})} \tilde{\Psi}_{t,W_\pi}(S^{(i)}_{1:t})\right]. \tag{21}$$

Next, we show that the rate of convergence suggested by (21) is in fact uniform over the whole class Π under appropriate bounded entropy conditions, thus enabling a regret bound for the oracle learner that optimizes (19). To do so, we introduce some more notation. Let $H = \{S_{1:T}, A_{1:T}\}$ be the entire history of a trajectory. Any policy π that is regular in the sense of Definition 1 can then be re-expressed as a mapping from H to a length $KT + 1$ vector of all zeros except for an indicator 1 at one position in the probability simplex, such that[10]

$$\pi(H) = \begin{cases} v_{K(\tau_\pi - 1) + W_\pi} & \text{if } \tau_\pi \leq T, \\ v_{KT+1} & \text{else,} \end{cases} \tag{22}$$

where $v_m \in \{0, 1\}^{KT+1}$ is the indicator vector with the mth position 1, and all others 0. Given this form, we note that we can re-express (20) as

$$\begin{aligned}\tilde{\Delta}(\pi, \pi') &= \frac{1}{n}\sum_{i=1}^{n} \langle \pi(H^{(i)}) - \pi'(H^{(i)}), \tilde{\Gamma}^{(i)} \rangle, \text{ where} \\ \tilde{\Gamma}^{(i)}_{K(t-1)+k} &= \sum_{t'=t}^{T} \frac{\mathbb{1}_{A^{(i)}_{1:(t'-1)}=0}}{\prod_{t''=1}^{t'-1} e_{t'',0}(S^{(i)}_{1:t''})} \tilde{\Psi}_{t,k}(S^{(i)}_{1:t'})\end{aligned} \tag{23}$$

for all $1 \leq t \leq T$ and $1 \leq k \leq K$, and $\tilde{\Gamma}^{(i)}_{KT+1} = 0$.

Given these preliminaries, let the Hamming distance between any two policies π, π' be

$$d_h(\pi, \pi') = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\pi(H^{(i)}) \neq \pi'(H^{(i)})}.$$

[10] All regular policies can be expressed in the form (22); however, we emphasize that the converse is not true: The form (22) does not ensure that π is $\mathcal{F}_t$ measurable.

Define the ε-Hamming covering number of Π as

$$N_{d_h}(\varepsilon, \Pi) = \sup\left\{N_{d_h}\left(\varepsilon, \Pi, \left\{H^{(1)}, \dots, H^{(n)}\right\}\right) \mid H^{(1)}, \dots, H^{(n)}\right\},$$

where $N_{d_h}\left(\varepsilon, \Pi, \left\{H^{(1)}, \dots, H^{(n)}\right\}\right)$ is the smallest number of policies $\pi^{(1)}, \pi^{(2)}, \dots, \in \Pi$ such that $\forall \pi \in \Pi, \exists \pi^{(i)}$ such that $d_h(\pi, \pi^{(i)}) \leq \varepsilon$. In our formal results, we control the complexity of the policy class Π in terms of its Hamming entropy.

Assumption 4. There exist constants $C, D \geq 0$ and $0 < \omega < 0.5$ such that, for all $0 < \varepsilon < 1$, $N_{d_h}(\varepsilon, \Pi) \leq C \exp(D(\frac{1}{\varepsilon})^{\omega})$.

Whenever Assumption 4 holds, we can use the argument from Lemma 2 of Zhou, Athey, and Wager (2018) to show the rate of convergence in (21) holds uniformly over the whole class Π for the oracle estimator $\tilde{\Delta}(\pi, \pi')$. The bounds below depend on the complexity of the class Π via

$$\kappa(\Pi) = \int_0^1 \sqrt{\log N_{d_h}(\varepsilon^2, \Pi)}\, d\varepsilon, \tag{24}$$

which is always finite under Assumption 4.

Example 2 (The class of linear thresholding policies). In the case of linear thresholding policies with binary actions $|\mathcal{A}| = 2$, that is, $\{\pi \in \Pi : \tau_\pi = \min(t : \theta^\top S_{1:t} > 0)\}$ where $\theta \in \mathbb{R}^d$, we note that by Haussler (1995), the covering number of a policy class for single-step decision-making is bounded by $N_{L_1(\mathbb{P}_n)}(\varepsilon, \Pi_t) \leq cVC(\Pi_t) \exp^{VC(\Pi_t)} (1/\varepsilon)^{VC(\Pi_t)}$ where $VC(\Pi_t)$ is the VC dimension of Π_t, the linear thresholding policy class at time t, and c is some numerical constant. Thus, with a different constant c, $N_{L_1(\mathbb{P}_n)}(\varepsilon, \Pi_t) \leq cde^d(1/\varepsilon)^d$. By taking a Cartesian product of the covering at each timestep and with a union bound on the error incurred at each timestep, we achieve a strict upperbound on $N_{d_h}(\varepsilon, \Pi) < cd^T e^{dT} (T/\varepsilon)^{dT}$ for a (again different) constant c, and so $\kappa(\Pi) < \sqrt{cdT \log(T)}$.

Lemma 5. Under Assumptions 1–4 and assuming $|Y| \leq M$ for some constant M almost surely, for any $\delta, c > 0$, there exists $0 < \varepsilon_0(\delta, c) < \infty$ and universal constants $0 < c_1, c_2 < \infty$ such that for all $\varepsilon < \varepsilon_0(\delta, c)$, if we collect at least $n(\varepsilon, \delta)$ samples, with

$$n(\varepsilon, \delta) = \frac{1}{\varepsilon^2}\left(c + \sqrt{V^*}\left(c_1\kappa(\Pi) + c_2 + \sqrt{2\log\left(\frac{1}{\delta}\right)}\right)\right)^2, \tag{25}$$

where $V^* = \sup_{\pi, \pi' \in \Pi} \mathbb{E}\left[\langle \pi(H^{(i)}) - \pi'(H^{(i)}), \tilde{\Gamma}^{(i)} \rangle^2\right]$, then, with probability at least $1 - 2\delta$,

$$\sup_{\pi, \pi' \in \Pi} \left|\tilde{\Delta}(\pi, \pi') - \Delta(\pi, \pi')\right| \leq \varepsilon, \tag{26}$$

and, moreover, letting $\tilde{\pi} = \operatorname{argmax}\{\tilde{\Delta}(\pi, \mathbf{0}) : \pi \in \Pi\}$ be the policy learned by optimizing the oracle objective (19), we have with probability at least $1 - 2\delta$, $R(\tilde{\pi}) \leq \varepsilon$.

Our goal is to get a comparable regret bound using the feasible estimator from (14) in Algorithm 1 that uses estimated nuisance components by coupling the feasible value estimates with the oracle ones. We establish our coupling result in terms of rates of convergence on the nuisance components, as follows.

Assumption 5. We work with a sequence of problems and estimators such that $\hat{\mu}^{-q(i)}_{\text{now},k}, \hat{\mu}^{-q(i)}_{\text{next},k}, \hat{e}^{-q(i)}_{t,k}$, satisfy for some universal constants C_μ, C_e, κ_μ, κ_e,

$$\sup_{k,t} \mathbb{E}\left[\left(\hat{\mu}^{-q(i)}_{\text{now},k}(S_{1:t}, t) - \mu_{\text{now},k}(S_{1:t}, t)\right)^2\right] \leq C_\mu n^{-2\kappa_\mu}, \tag{27}$$

$$\sup_{k,t} \mathbb{E}\left[\left(\hat{\mu}^{-q(i)}_{\text{next},k}(S_{1:t}, t) - \mu_{\text{next},k}(S_{1:t}, t)\right)^2\right] \leq C_\mu n^{-2\kappa_\mu}, \tag{28}$$

$$\sup_{k,t} \mathbb{E}\left[\left(\frac{1}{\hat{e}^{-q(i)}_{t,k}(S_{1:t})} - \frac{1}{e_{t,k}(S_{1:t})}\right)^2\right] \leq C_e n^{-2\kappa_e}, \tag{29}$$

and furthermore $\hat{e}^{-q(i)}_{t,k}(S_{1:t})$ is uniformly consistent,

$$\sup_{t, k, s \in \mathcal{S}^T} \left|\hat{e}^{-q(i)}_{t,k}(s_{1:t}) - e_{t,k}(s_{1:t})\right| \to_p 0. \tag{30}$$

Moreover, motivated by the observation that treatment effects are weak relative to the available sample size in many problems of interest, we allow for problem sequences where treatment effects can shrink with sample size n. In regimes where treatment effects stay constant when the sample size grows, super-efficiency phenomena are unavoidable; see Luedtke and Chambaz (2017) for a formal statement with static decision rules. To consider other such settings, first recall the definition of $\delta_{\text{local},k}$ defined in (13). We further define $\delta^+_{\text{local},k}(S_{1:t+1}, t) = \mu_{\text{now},k}(s_{1:t+1}, t+1) - \mu_{\text{next},k}(s_{1:t}, t)$.[11]

Assumption 6. For some universal constants C_δ, κ_δ, C_γ, κ_γ,

$$\sup_{t,k} \mathbb{E}\left[\delta_{\text{local},k}(S_{1:t}, t)^2\right] \leq C_\delta n^{-2\kappa_\delta}, \tag{31}$$

$$\sup_{t,k} \mathbb{E}\left[\delta^+_{\text{local},k}(S_{1:t+1}, t)^2\right] \leq C_\gamma n^{-2\kappa_\gamma}. \tag{32}$$

Lemma 6. Suppose that Assumptions 1–6 hold and assume $|Y| \leq M$ for some constant M almost surely. Then, for any $\delta > 0$, there exists $0 < \varepsilon_0(\delta, \eta, T) < \infty$ such that for all $\varepsilon < \varepsilon_0(\delta, \eta, T)$, with probability at least $1 - 3\delta$,

$$\sup_{\pi, \pi'} \left|\hat{\Delta}(\pi, \pi') - \tilde{\Delta}(\pi, \pi')\right| \leq \varepsilon,$$

provided we collect at least $n_0(\varepsilon, \delta)$ samples, where

$$n_0(\varepsilon, \delta) = \left(C(\delta) K T^2 \varepsilon^{-1}\right)^{1/\min\{1/2+\kappa_e, 1/2+\kappa_\mu, \kappa_e+\kappa_\mu, \kappa_e+\kappa_\delta, \kappa_e+\kappa_\gamma\}},$$

where $C(\delta)$ only depends on the constants used in Assumptions 3, 5, and 6.

Combining the above with Lemma 5, we immediately have the following finite-sample bound for the regret on the feasible estimator.

[11] In deterministic systems, this difference is always identically zero, but in stochastic settings the two quantities will generally be different. Intriguingly, related quantities (the expected temporal-difference error, and the variance of the value of next state) have been observed to play important roles in online reinforcement learning regret bounds (see, e.g., Zanette and Brunskill 2019) as well.

Theorem 7. Let $\hat{\pi} = \operatorname{argmax}\{\hat{\Delta}(\pi, \mathbf{0}) : \pi \in \Pi\}$ be the policy learned by optimizing the feasible objective (14). Suppose Assumptions 1–6 and assume $|Y| \leq M$ for some constant M almost surely. Then, for any $\delta > 0$, there exist $0 < \varepsilon_0(\delta, \eta, T) < \infty$ such that the following statement holds for all $\varepsilon < \varepsilon_0(\delta, \eta, T)$: If we collect at least $n(\varepsilon, \delta)$ samples, with

$$n(\varepsilon, \delta) = \max\left\{ \frac{1}{\varepsilon^2}\left(c + \sqrt{V^*}\left(c_1\kappa(\Pi) + c_2 + \sqrt{2\log\left(\frac{1}{\delta}\right)}\right)\right)^2, \right. \\ \left. n_0(\varepsilon, \delta) \right\}, \tag{33}$$

$V^* = \sup_{\pi,\pi' \in \Pi} \mathbb{E}\left[\langle \pi(H^{(i)}) - \pi'(H^{(i)}), \tilde{\Gamma}^{(i)}\rangle^2\right]$, and $n_0(\varepsilon, \delta)$ as defined in Lemma 6, then with probability at least $1 - 5\delta$

$$\sup_{\pi,\pi' \in \Pi} \left| \hat{\Delta}(\pi, \pi') - \Delta(\pi, \pi') \right| \leq 2\varepsilon,$$

and in particular $R(\hat{\pi}) \leq 2\varepsilon$.

We obtain the following corollary if we assume specific learning rates on the nuisance components and the signal strength.

Corollary 8. Assume $\kappa_\mu > 0$, $\kappa_e > 0$, $\kappa_e + \kappa_\mu > \frac{1}{2}$, $\kappa_e + \kappa_\gamma > \frac{1}{2}$, $\kappa_e + \kappa_\delta > \frac{1}{2}$. Suppose Assumptions 1–6 hold and assume $|Y| \leq M$ for some constant M almost surely. Then, for any $\delta > 0$, there exist $0 < \varepsilon_0(\delta, \eta, T) < \infty$ such that the following holds for all $\varepsilon < \varepsilon_0(\delta, \eta, T)$: If we collect at least $n(\varepsilon, \delta)$ samples, with

$$n(\varepsilon, \delta) = \frac{1}{\varepsilon^2}\left(c + \sqrt{V^*}\left(c_1\kappa(\Pi) + c_2 + \sqrt{2\log\left(\frac{1}{\delta}\right)}\right)\right)^2, \tag{34}$$

and $V^* = \sup_{\pi,\pi' \in \Pi} \mathbb{E}\left[\langle \pi(H^{(i)}) - \pi'(H^{(i)}), \tilde{\Gamma}^{(i)}\rangle^2\right]$, then with probability at least $1 - 5\delta$

$$\sup_{\pi,\pi' \in \Pi} \left| \hat{\Delta}(\pi, \pi') - \Delta(\pi, \pi') \right| \leq 2\varepsilon,$$

and in particular $R(\hat{\pi}) \leq 2\varepsilon$.

Our result above can be interpreted in several different regimes. First, we note that we can reach the optimal sample complexity $n \sim \varepsilon^{-2}$ if either (a) the treatment propensities $e_{t,k}$ are known and we can consistently estimate $\mu_{\text{now},k}$ and $\mu_{\text{next},k}$; or (b) the signal size of the advantages is null (i.e., $\mu_{\text{now},k}(S_{1:t}, t) - \mu_{\text{next},k}(S_{1:t}, t) = 0$) or is weak (in the sense that $\kappa_\delta > 0$ and $e_{t,0}$ can be learned at a rate such that $\kappa_\delta + \kappa_e > 1/2$, etc.), and similarly the stochastic fluctuations are weak (in the sense that $\kappa_\gamma > 0$ and $e_{t,0}$ can be learned at a rate such that $\kappa_\gamma + \kappa_e > 1/2$, etc.), and we can consistently estimate $\mu_{\text{now},k}$, $\mu_{\text{next},k}$ and $e_{t,k}$ such that $\kappa_e + \kappa_\mu > 1/2$.

Conversely, if the treatment effects are of a fixed size (i.e., $\kappa_\delta = 0$), and we do not know the treatment propensities $e_{t,k}$ a priori, then we pay a price for not being robust to the change of measure from Lemma 1 to Lemma 2, and we no longer achieve the optimal rate. The terms that hurt us are due to error terms that decay as $n^{-(\kappa_e + \kappa_\delta)}$ which arises from the interaction of how we use inverse propensity weighting for the treatment starting probabilities and the signal size of the advantages, and ones that decays as $n^{-(\kappa_e + \kappa_\gamma)}$ which are similarly due to stochastic fluctuations in the value of starting treatment. If advantages are small, this will not matter for smaller target error rates ε, but requires a bigger sample size when we aim for very small ε.

4. ADR With a Terminal State

So far, we have focused our analysis on when-to-stop problems in settings characterized by overlap (Assumption 3), that is, where the sampling policy can start treatment in any state with a nonzero policy, and have assumed that we want to learn a regular policy in the sense of Definition 1, that is, one that never stops prescribing treatment once it has started to do so. In many applications of interest, however, a patient may enter a terminal state in which treatment becomes impossible—for example, a patient may leave the study or die. The existence of such a terminal state contradicts the assumptions made above: There is no overlap in the terminal state (because treatment can never start there), and a policy that respects the terminal state may not be regular (because the policy must stop prescribing treatment once the patient enters the terminal state).

The goal of this section is to briefly discuss methodological extensions to ADR that are required in the presence of a terminal state. Algorithm 2 provides pseudocode for our ADR policy optimization approach with terminal states.

To do so, we start by adapting Definition 1 and Assumption 3 to this setting.

Definition 3 (Terminal state). A state $\Phi \in \mathcal{S}$ is terminal if, whenever $S_t = \Phi$, then also $S_{t'} = \Phi$ for all $t' > t$. Furthermore, we assume that once a patient enters a terminal state, we can assess their final outcome, that is, there exists a set of known functions[12] H_t such that $Y = H_t(S_{1:t})$ whenever $S_{t+1} = \Phi$.

Definition 1b (Regular policy with terminal state). A regular when-to-treat policy π that respects the terminal state Φ is determined by an $\mathcal{F}_t$-measurable stopping time τ_π and an associated $\mathcal{F}_{\tau_\pi}$-measurable decision variable $W_\pi \in \{1, \ldots, K\}$ as follows[13]: For each time $t = 1, \ldots, T$, if $S_t = \Phi$ then $\pi_t(S_{1:t}, A_{1:(t-1)}) = 0$. Otherwise, if $A_{t-1} \neq 0$ then $\pi_t(S_{1:t}, A_{1:(t-1)}) = A_{t-1}$, else if $t \geq \tau_\pi$ then $\pi_t(S_{1:t}, A_{1:(t-1)}) = W_\pi$. If none of the above conditions apply, then $\pi_t(S_{1:t}, A_{1:(t-1)}) = 0$.

Assumption 3b (Overlap with terminal state). There are constants η, $\eta_0 > 0$ as well as a terminal state Φ such that, for all $t = 1, \ldots, T$ and $s_{1:t} \in \mathcal{S}^t$, the following hold. If $s_t = \Phi$, then $e_{t,a}(s_{1:t}) = 0$ for all $a \in \mathcal{A} \setminus \{0\}$ and $e_{t,0}(s_{1:t}) = 1$; else, $e_{t,a}(s_{1:t}) > \eta/T$ for all $a \in \mathcal{A} \setminus \{0\}$ and $e_{t,0}(s_{1:t}) > 1 - \eta_0/T$.

[12] For example, if Y is survival time, then one can use $H_t(S_{1:t}) = \sup\left\{t' : S_{t'} \neq \Phi, t' \leq t\right\}$.

[13] Note that the policies specified here are still when-to-start policies, that is, if they have actually started treatment then they never stop (even if the patient enters a terminal state). One could also choose to make π stop treatment once the patient enters a terminal state. However, from a statistical perspective, this makes no difference: All that matters is that the standard of care is a deterministic function of state once treatment has started.

In the presence of a terminal state, the main modification we need to make to ADR is that the conditional expectation $\mu_{\text{next},k}(s_{1:t}, t)$ as defined in (8) no longer matches the Q-function that arises in (10) in the proof of Lemma 1, and so we need to adapt our statement of this result. The proof of the following lemma is included in Appendix A in the supplementary materials.

Lemma 9. Under Assumptions 1 and 2, let Φ be the terminal state, and let π be a regular when-to-treat policy that respects Φ in the sense of Definition 1b. Then

$$\Delta(\pi, \mathbf{0}) = \mathbb{E}_{\mathbf{0}}\left[\sum_{t=\tau_\pi}^{T} \mathbb{1}_{S_t \neq \Phi}\left(\mu_{\text{now},W_\pi}(S_{1:t}, t) - \mu^{\Phi}_{\text{next},W_\pi}(S_{1:t}, t)\right)\right], \tag{35}$$

where $\mathbb{E}_{\mathbf{0}}$ samples trajectories under a never-treating policy, τ_π is the time at which π starts treating and W_π is the treatment chosen at that time, and

$$\begin{aligned} &\mu^{\Phi}_{\text{next},k}(S_{1:t}, t) \\ &\quad = \mathbb{P}\left[S_{t+1} \neq \Phi \,\middle|\, S_{1:t},\, A_{1:t} = \mathbf{0}\right] \mathbb{E}\Big[\mu_{\text{now},k}(S_{1:t+1},\, t+1) \,\Big|\, S_{1:t}, \\ &\qquad A_{1:t} = \mathbf{0},\, S_{t+1} \neq \Phi\Big] \\ &\qquad + \mathbb{P}\left[S_{t+1} = \Phi \,\middle|\, S_{1:t},\, A_{1:t} = \mathbf{0}\right] H_t(S_{1:t}). \end{aligned} \tag{36}$$

Next, the following result is a direct consequence of Lemma 9; its proof is a direct analogue to that of Lemma 2 and thus omitted.

Lemma 10. In the setting of Lemma 9 and under Assumptions 1, 2, and 3b,

$$\Delta(\pi, \mathbf{0}) = \mathbb{E}\Bigg[\sum_{t=\tau_\pi}^{T} \mathbb{1}_{S_t \neq \Phi} \frac{\mathbb{1}_{A_{1:t-1}=0}}{\prod_{t'=1}^{t-1} e_{t',0}(S_{1:t'})}\Big(\mu_{\text{now},W_\pi}(S_{1:t}, t) \tag{37}$$

$$- \mu^{\Phi}_{\text{next},W_\pi}(S_{1:t}, t)\Big)\Bigg]. \tag{38}$$

We detail a candidate estimator based on this result as Algorithm 2. For notational convenience, we denote terminating probabilities by $\rho(S_{1:t}) = \mathbb{P}\left[S_{t+1} = \Phi \,\middle|\, S_{1:t},\, A_{1:t} = \mathbf{0}\right]$, and write $U(S_{1:t},\, \Phi) = \mathbb{E}\left[\frac{\mathbb{1}_{A_{t+1}=k}}{e_{t+1,k}(S_{1:t+1})} Y \,\middle|\, S_{1:t},\, A_{1:t} = \mathbf{0},\, S_{t+1} \neq \Phi\right]$. We note that the proposed ADR estimator extended to terminal states is robust toward errors in estimating the regression outcome functions $\mu_{\text{now},k}$ and $U(\cdot, \Phi)$) but we do not correct for the estimation bias in estimating the terminating probabilities $\rho(\cdot)$. We leave it to future work to develop robust methods that are also robust against terminating probability estimates.

Algorithm 2: Advantage doubly robust estimator with terminal state

```
1 Estimate the outcome models $\mu_{\text{now},k}(\cdot)$, $U(\cdot, \Phi)$,
  terminating propensities $\rho(\cdot)$ as well as treatment
  propensities $e_{t,a}(s_{1:t})$ with cross-fitting using any
  supervised learning method tuned for prediction accuracy.
2 Given these nuisance component estimates, we construct
  value estimates
      $$\hat{\Delta}(\pi, \mathbf{0}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{T} \mathbb{1}_{S_t \neq \Phi}\mathbb{1}_{t \geq \tau_\pi^{(i)}} \frac{\mathbb{1}_{A^{(i)}_{1:t-1}=0}}{\prod_{t'=1}^{t-1} \hat{e}^{-q(i)}_{t',0}(S^{(i)}_{1:t'})} \hat{\Psi}^{\Phi}_{t,W_\pi}(S^{(i)}_{1:t}) \tag{40}$$
3 for each policy $\pi \in \Pi$, where the relevant DR score is
      $$\begin{aligned} \hat{\Psi}^{\Phi}_{t,k}(S^{(i)}_{1:t}) = {} & \hat{\mu}^{-q(i)}_{\text{now},k}(S^{(i)}_{1:t}, t) - \hat{\mu}^{\Phi}_{\text{next},k}(S^{(i)}_{1:t}, t)^{-q(i)} \\ & + \mathbb{1}_{A^{(i)}_t = k} \frac{Y^{(i)} - \hat{\mu}^{-q(i)}_{\text{now},k}(S^{(i)}_{1:t}, t)}{\hat{e}^{-q(i)}_{t,k}(S^{(i)}_{1:t})} \\ & - \mathbb{1}_{A^{(i)}_t = 0}\mathbb{1}_{A^{(i)}_{t+1} = k} \frac{Y^{(i)} - \hat{U}^{-q(i)}(S^{(i)}_{1:t}, \Phi)}{\hat{e}^{-q(i)}_{t,0}(S^{(i)}_{1:t})\hat{e}^{-q(i)}_{t+1,k}(S^{(i)}_{1:t+1})}, \end{aligned} \tag{41}$$
  and $\hat{\mu}^{\Phi}_{\text{next},k}(S^{(i)}_{1:t}, t)^{-q(i)} =$
  $(1 - \hat{\rho}^{-q(i)}(S^{(i)}_{1:t}))\hat{U}^{-q(i)}(S^{(i)}_{1:t}, \Phi) + \hat{\rho}^{-q(i)}(S^{(i)}_{1:t})H_t(S^{(i)}_{1:t})$
4 Learn the optimal policy by setting
  $\hat{\pi} = \text{argmax}_{\pi \in \Pi}\hat{\Delta}(\pi, \mathbf{0})$.
```

Finally, in analogy to (16), it is convenient to re-express $\mu_{\text{next},k}(S_{1:t}.t)^{\Phi}$ via inverse-propensity weighting for purpose of estimating it,

$$\begin{aligned} &\mu^{\Phi}_{\text{next},k}(S_{1:t}, t) \\ &\quad = \mathbb{P}\left[S_{t+1} \neq \Phi \,\middle|\, S_{1:t},\, A_{1:t} = \mathbf{0}\right] \mathbb{E}\Bigg[\frac{\mathbb{1}_{A_{t+1}=k}}{e_{t+1,k}(S_{1:t+1})} Y \,\Bigg|\, S_{1:t}, \\ &\quad A_{1:t} = \mathbf{0},\, S_{t+1} \neq \Phi\Bigg] \\ &\qquad + \mathbb{P}\left[S_{t+1} = \Phi \,\middle|\, S_{1:t},\, A_{1:t} = \mathbf{0}\right] H_t(S_{1:t}). \end{aligned} \tag{39}$$

Furthermore, a weighted regression expression analogous to (17) also holds.

5. Experiments

To assess the practical performance of our proposed method, we consider two different simulation studies. In the first simulation, we consider the optimal stopping case in which the treatment decision is binary and the treatment assignment propensities are not known a priori. In the second simulation, we want to learn when to start which treatment. There are multiple treatment options and patients can be censored due to death, and the data are generated from a randomized control trial with known treatment assignment propensities. This second study helps to capture settings motivated by clinical trials.

In both settings, we consider linear thresholding policy rules for simplicity and due to their interpretability. In our implementation, we use the normalized variant of the IPW estimator

$\hat{V}_\pi^{\text{WIPW}}$ as presented in Section 2.2. For the first setup that does not involve survival censoring, we use a correspondingly normalized ADR estimator $\hat{\Delta}^{\text{W}}$ in Step 2 of Algorithm 1:

$$\hat{\Delta}^{\text{W}}(\pi,\mathbf{0}) = \sum_{t=1}^{T} \frac{\sum_{i=1}^{n} \frac{\mathbb{1}_{A^{(i)}_{1:t-1}=0}}{\prod_{t'=1}^{t-1} \hat{e}^{-q(i)}_{t',0}(S^{(i)}_{1:t'})} \mathbb{1}_{t\geq\tau^{(i)}_\pi} \left(\hat{\mu}^{-q(i)}_{\text{now},W_\pi}(S^{(i)}_{1:t},t) - \hat{\mu}^{-q(i)}_{\text{next},W_\pi}(S^{(i)}_{1:t},t)\right)}{\sum_{i=1}^{n} \frac{\mathbb{1}_{A^{(i)}_{1:t-1}=0}}{\prod_{t'=1}^{t-1} \hat{e}^{-q(i)}_{t',0}(S^{(i)}_{1:t'})}}$$

$$+ \sum_{t=1}^{T} \frac{\sum_{i=1}^{n} \frac{\mathbb{1}_{A^{(i)}_{1:t-1}=0}\mathbb{1}_{A^{(i)}_{t}=W_\pi}}{\prod_{t'=1}^{t-1} \hat{e}^{-q(i)}_{t',0}(S^{(i)}_{1:t'})\hat{e}^{-q(i)}_{t,W_\pi}(S^{(i)}_{1:t})} \mathbb{1}_{t\geq\tau^{(i)}_\pi} \left(Y^{(i)} - \hat{\mu}^{-q(i)}_{\text{now},W_\pi}(S^{(i)}_{1:t},t)\right)}{\sum_{i=1}^{n} \frac{\mathbb{1}_{A^{(i)}_{1:t-1}=0}\mathbb{1}_{A^{(i)}_{t}=W_\pi}}{\prod_{t'=1}^{t-1} \hat{e}^{-q(i)}_{t',0}(S_{1:t'})\hat{e}^{-q(i)}_{t,W_\pi}(S^{(i)}_{1:t})} \mathbb{1}_{t\geq\tau^{(i)}_\pi} + \sum_{i=1}^{n} \frac{\mathbb{1}_{A^{(i)}_{1:t-1}=0}}{\prod_{t'=1}^{t-1} \hat{e}^{-q(i)}_{t',0}(S_{1:t'})}(1-\mathbb{1}_{t\geq\tau^{(i)}_\pi})}$$

$$- \sum_{t=1}^{T} \frac{\sum_{i=1}^{n} \frac{\mathbb{1}_{A^{(i)}_{1:t}=0}\mathbb{1}_{A^{(i)}_{t+1}=W_\pi}}{\prod_{t'=1}^{t} \hat{e}^{-q(i)}_{t',0}(S_{1:t'})\hat{e}^{-q(i)}_{t+1,W_\pi}(S^{(i)}_{1:t+1})} \mathbb{1}_{t\geq\tau^{(i)}_\pi} \left(Y^{(i)} - \hat{\mu}^{-q(i)}_{\text{next},W_\pi}(S^{(i)}_{1:t},t)\right)}{\frac{\mathbb{1}_{A^{(i)}_{1:t}=0}\mathbb{1}_{A^{(i)}_{t+1}=W_\pi}}{\prod_{t'=1}^{t} \hat{e}^{-q(i)}_{t',0}(S_{1:t'})\hat{e}^{-q(i)}_{t+1,W_\pi}(S^{(i)}_{1:t+1})} \mathbb{1}_{t\geq\tau^{(i)}_\pi} + \frac{\mathbb{1}_{A^{(i)}_{1:t-1}=0}}{\prod_{t'=1}^{t-1} \hat{e}^{-q(i)}_{t',0}(S_{1:t'})}(1-\mathbb{1}_{t\geq\tau^{(i)}_\pi})}.$$

For simplicity, we will refer to them as the IPW (baseline) and ADR (our estimator), respectively, in this section. We have a similar weighted form for the ADR estimator with terminal states that we use in experiments in the second simulation study. We include that in Appendix B in the supplementary materials.

In addition to IPW, we also consider fitted-Q iteration as a baseline method for policy learning. The variant of fitted-Q iteration we implement follows the batch Q-learning algorithm as described in Murphy (2005) for solving the optimal Q function at each timestep: At each $t = T, T-1, \ldots, 1$, we solve[14]

$$\hat{Q}^*_t(\cdot,\cdot) = \text{argmin}_{Q_t} \frac{1}{n}\sum_{i=1}^{n}\left(\max_{a_{t+1}} \hat{Q}^*_{t+1}(S^{(i)}_{1:(t+1)}, \{A^{(i)}_{1:t}, a_{t+1}\}) - Q_t(S^{(i)}_{1:t}, A^{(i)}_{1:t})\right)^2, \tag{42}$$

where we let $\hat{Q}^*_{T+1} = Y^{(i)}$. We note that fitted-Q iteration is an iterative backward-regression based algorithm targeted at learning the optimal unrestricted policy, whereas our goal is to learn the best in-class policy given a user-defined policy class. However, while fitted-Q aims to perform a different task than us, it is still of interest to compare the regret achieved by both methods. We use the shorthand *Q-Opt* to refer to this method.

Another variant of the fitted-Q method evaluates a given policy π instead of learning the best policy (see, e.g., Le, Voloshin, and Yue 2019). Instead of taking the max operator above, a_{t+1} is chosen according to the policy π. We call the latter fitted-Q for evaluation and use *Q-Eval* as a shorthand accordingly.

5.1. *Binary Treatment Choices in an Observational Study*

Our first simulation is motivated by a setting where we track a health metric and get a reward if the health metric is above a threshold at $T = 10$. The treatment provides a positive nudge to the health metric at a cost. We start with treatment on, and need to choose when to stop to minimize cost while trying to keep the health metric stay above the threshold. The data are generated via the following hidden Markov process, where X_t is unobserved:

$$X_1 \sim \mathcal{N}(0,\sigma^2),$$

$$X_{t+1} \,\big|\, X_t, A_t \sim \mathbb{1}_{X_t\geq -0.5}\mathcal{N}\left(X_t + \frac{1}{1+e^{0.3X_t}}A_t, \frac{\sigma^2}{2T}\right) + \mathbb{1}_{X_t<-0.5}X_t,\ S_t \,\big|\, X_t \sim \mathcal{N}(X_t,\nu^2),$$

$$Y = \beta\mathbb{1}_{S_{T+1}>0} - \frac{1}{T}\sum_{t=1}^{T}A_t,$$

with the stopping action $A_t \,\big|\, X_t \sim \text{Bernoulli}(1 - 1/(1 + e^{-(X_t-1.5)} - e^{-(t-3)}))$. We note that Y is the final outcome we would like to maximize. We also do not assume Markovian structure and only get to observe S_t, which is a noisy version of the underlying state X_t.

In our implementation, both the propensity and outcome regressions only use the current state and action information as opposed to the full history even though the underlying dynamic is not Markovian.[15] We parameterize the policy class of interest by $[\theta_1,\theta_2,\theta_3]$ and define each policy to be a linear thresholding rule $\theta_1 S_t \geq \theta_2 t + \theta_3$ such that whenever this holds, we stop the treatment. We then perform a grid search over a range of values for the policy parameters, with the grid specified in Appendix B in the supplementary materials.

For each of the parameter combinations, we run ADR and baseline IPW to estimate the value of the corresponding policy. The average mean-squared error (MSE) of each of the policy values across all policies in the policy class is then computed against an oracle evaluation by using a Monte Carlo rollout of the policy using the underlying transition dynamics averaged across 20,000 times. We vary β, σ, and the observation noise ν and compute the regret and the mean-squared error of policy value estimates (averaged across all policies in the policy class).

ADR shows a clear advantage in both regret and learning the correct value of policies across varying values of σ, β, and ν. We present the tables of raw results for in Tables 2 and 1 in Appendix B in the supplementary materials. We present one representative illustration in Figure 1, where we have used

[14] For this purpose, we estimate propensities and conditional response surfaces using regression forests as implemented in `grf` (Athey, Tibshirani, and Wager 2019). For tractability, we do not consider history when learning these regression; rather, we only use current state as covariates in each time step.

[15] In other words, neither out propensity models nor conditional response models are well specified because we do not use covariates that capture lagged states. Thus, this setting can be seen as a test case for the value of the robust scoring method in ADR.

Table 1. Detailed numerical results in the binary-action setup with $\beta = 1$; details see caption of Table 2.

n	ν	β	σ	ADR	IPW	Q-Opt	Oracle	MSE:ADR	MSE:IPW
250	0	1	1	7.65e-01	7.21e-01	**8.21e-01**	8.03e-01	**1.56e-02**	3.32e-01
250	0	1	3	8.16e-01	7.11e-01	**8.17e-01**	8.46e-01	**3.20e-02**	2.43e-01
250	0.5	1	1	**7.48e-01**	7.25e-01	7.38e-01	7.79e-01	**1.07e-01**	3.28e-01
250	0.5	1	3	**8.18e-01**	7.15e-01	8.11e-01	8.49e-01	**6.97e-02**	2.55e-01
500	0	1	1	7.86e-01	7.50e-01	**8.36e-01**	8.03e-01	**4.70e-03**	3.16e-01
500	0	1	3	**8.39e-01**	7.46e-01	8.26e-01	8.46e-01	**1.27e-02**	2.05e-01
500	0.5	1	1	**7.56e-01**	7.28e-01	7.47e-01	7.79e-01	**7.20e-02**	3.21e-01
500	0.5	1	3	**8.37e-01**	7.52e-01	8.17e-01	8.49e-01	**1.98e-02**	2.31e-01
1000	0	1	1	7.88e-01	7.74e-01	**8.46e-01**	8.03e-01	**3.09e-03**	2.87e-01
1000	0	1	3	**8.42e-01**	7.81e-01	8.31e-01	8.46e-01	**9.20e-03**	2.12e-01
1000	0.5	1	1	**7.63e-01**	7.41e-01	7.49e-01	7.79e-01	**6.16e-02**	2.82e-01
1000	0.5	1	3	**8.42e-01**	7.81e-01	8.16e-01	8.49e-01	**2.87e-02**	2.03e-01
5000	0	1	1	7.97e-01	7.94e-01	**8.54e-01**	8.03e-01	**1.62e-03**	2.05e-01
5000	0	1	3	**8.46e-01**	8.32e-01	8.39e-01	8.46e-01	**4.92e-03**	1.42e-01
5000	0.5	1	1	**7.68e-01**	7.60e-01	7.53e-01	7.79e-01	**2.46e-02**	1.58e-01
5000	0.5	1	3	**8.49e-01**	8.19e-01	8.22e-01	8.49e-01	**1.37e-02**	1.39e-01
10,000	0	1	1	8.00e-01	7.98e-01	**8.54e-01**	8.03e-01	**9.82e-04**	1.69e-01
10,000	0	1	3	**8.46e-01**	8.40e-01	8.42e-01	8.46e-01	**4.53e-03**	1.52e-01
10,000	0.5	1	1	**7.69e-01**	7.64e-01	7.52e-01	7.79e-01	**1.45e-02**	9.67e-02
10,000	0.5	1	3	**8.49e-01**	8.33e-01	8.23e-01	8.49e-01	**1.10e-02**	1.53e-01
20,000	0	1	1	8.00e-01	8.01e-01	**8.54e-01**	8.03e-01	**3.54e-04**	9.76e-02
20,000	0	1	3	**8.46e-01**	8.42e-01	8.42e-01	8.46e-01	**3.23e-03**	1.26e-01
20,000	0.5	1	1	**7.69e-01**	7.67e-01	7.54e-01	7.79e-01	**7.74e-03**	5.15e-02
20,000	0.5	1	3	**8.49e-01**	8.42e-01	8.21e-01	8.49e-01	**1.01e-02**	1.42e-01
30,000	0	1	1	8.00e-01	8.00e-01	**8.54e-01**	8.03e-01	**3.22e-04**	9.42e-02
30,000	0	1	3	**8.46e-01**	8.43e-01	8.41e-01	8.46e-01	**2.85e-03**	1.19e-01
30,000	0.5	1	1	**7.70e-01**	7.68e-01	7.54e-01	7.79e-01	**7.99e-03**	5.27e-02
30,000	0.5	1	3	**8.49e-01**	8.44e-01	8.22e-01	8.49e-01	**9.88e-03**	1.11e-01

Table 2. Detailed numerical results in the binary-action setup with $\beta = 0.5$.

n	ν	β	σ	ADR	IPW	Q-Opt	Oracle	MSE:ADR	MSE:IPW
250	0	0.5	1	8.39e-01	8.38e-01	**8.53e-01**	8.78e-01	**1.70e-02**	8.63e-02
250	0	0.5	3	**9.11e-01**	8.25e-01	8.79e-01	9.25e-01	**2.42e-02**	6.63e-02
250	0.5	0.5	1	8.32e-01	**8.33e-01**	8.02e-01	8.76e-01	**3.51e-02**	8.27e-02
250	0.5	0.5	3	**9.18e-01**	8.61e-01	8.78e-01	9.27e-01	**2.51e-02**	6.85e-02
500	0	0.5	1	**8.73e-01**	8.49e-01	8.71e-01	8.78e-01	**5.50e-03**	8.11e-02
500	0	0.5	3	**9.23e-01**	8.66e-01	8.79e-01	9.25e-01	**5.76e-03**	6.60e-02
500	0.5	0.5	1	**8.69e-01**	8.62e-01	8.11e-01	8.76e-01	**3.23e-02**	7.54e-02
500	0.5	0.5	3	**9.23e-01**	8.77e-01	8.84e-01	9.27e-01	**1.44e-02**	6.07e-02
1000	0	0.5	1	**8.78e-01**	8.71e-01	8.74e-01	8.78e-01	**4.50e-03**	8.06e-02
1000	0	0.5	3	**9.25e-01**	8.75e-01	8.87e-01	9.25e-01	**6.67e-03**	5.76e-02
1000	0.5	0.5	1	**8.73e-01**	8.70e-01	8.16e-01	8.76e-01	**2.25e-02**	7.19e-02
1000	0.5	0.5	3	**9.27e-01**	8.88e-01	8.84e-01	9.27e-01	**8.94e-03**	5.64e-02
5000	0	0.5	1	**8.81e-01**	8.79e-01	8.79e-01	8.78e-01	**1.29e-03**	5.56e-02
5000	0	0.5	3	**9.25e-01**	9.11e-01	8.98e-01	9.25e-01	**3.20e-03**	4.40e-02
5000	0.5	0.5	1	**8.78e-01**	8.75e-01	8.30e-01	8.76e-01	**8.48e-03**	5.35e-02
5000	0.5	0.5	3	**9.27e-01**	9.13e-01	8.91e-01	9.27e-01	**6.35e-03**	3.63e-02
10,000	0	0.5	1	**8.81e-01**	8.80e-01	8.80e-01	8.78e-01	**8.70e-04**	4.33e-02
10,000	0	0.5	3	**9.25e-01**	9.20e-01	8.99e-01	9.25e-01	**2.26e-03**	4.26e-02
10,000	0.5	0.5	1	**8.78e-01**	8.76e-01	8.33e-01	8.76e-01	**4.72e-03**	3.07e-02
10,000	0.5	0.5	3	**9.27e-01**	9.19e-01	8.93e-01	9.27e-01	**5.77e-03**	3.27e-02
20,000	0	0.5	1	**8.81e-01**	**8.81e-01**	8.80e-01	8.78e-01	**5.51e-04**	3.09e-02
20,000	0	0.5	3	**9.25e-01**	9.21e-01	9.01e-01	9.25e-01	**1.56e-03**	3.37e-02
20,000	0.5	0.5	1	**8.78e-01**	8.77e-01	8.36e-01	8.76e-01	**2.96e-03**	1.56e-02
20,000	0.5	0.5	3	**9.27e-01**	9.21e-01	8.95e-01	9.27e-01	**4.57e-03**	3.48e-02
30,000	0	0.5	1	**8.81e-01**	**8.81e-01**	8.80e-01	8.78e-01	**2.63e-04**	1.87e-02
30,000	0	0.5	3	**9.25e-01**	9.24e-01	9.02e-01	9.25e-01	**1.46e-03**	2.83e-02
30,000	0.5	0.5	1	**8.78e-01**	8.77e-01	8.36e-01	8.76e-01	**2.20e-03**	1.18e-02
30,000	0.5	0.5	3	**9.27e-01**	9.25e-01	8.93e-01	9.27e-01	**3.75e-03**	2.98e-02

NOTE: In the fifth to the eighth columns, we show the value of the best learned policy using ADR, weighted IPW, and *Q-Opt* against the value of the oracle (oracle) best policy in the prespecified policy class, with all value estimates evaluated using a Monte Carlo rollout with 20,000 repeats. In the right two columns, we show the mean-squared error of the value estimates averaged across all policies in the policy class. Results are averaged across 50 runs and rounded to two decimal places. Numbers listed are accurate up to the second displaying digit due to sampling errors.

$\sigma = 1$, $\beta = 1$, and $\nu = 0.5$. We compare the performance of ADR against IPW and *Q-Opt* with varying numbers of offline trajectories. IPW and ADR first evaluate the values of the policies in the policy class, and so we plot the MSE of their policy estimates averaged across all policies in the policy class in the right plot; it is not applicable for *Q-Opt* which seeks to learn the optimal policy directly.

To gain further insight into the poor performance of the Q-learning baseline, Figure 2 decision rules learned by both ADR and *Q-Opt* in a single simulation realization. Specifically,

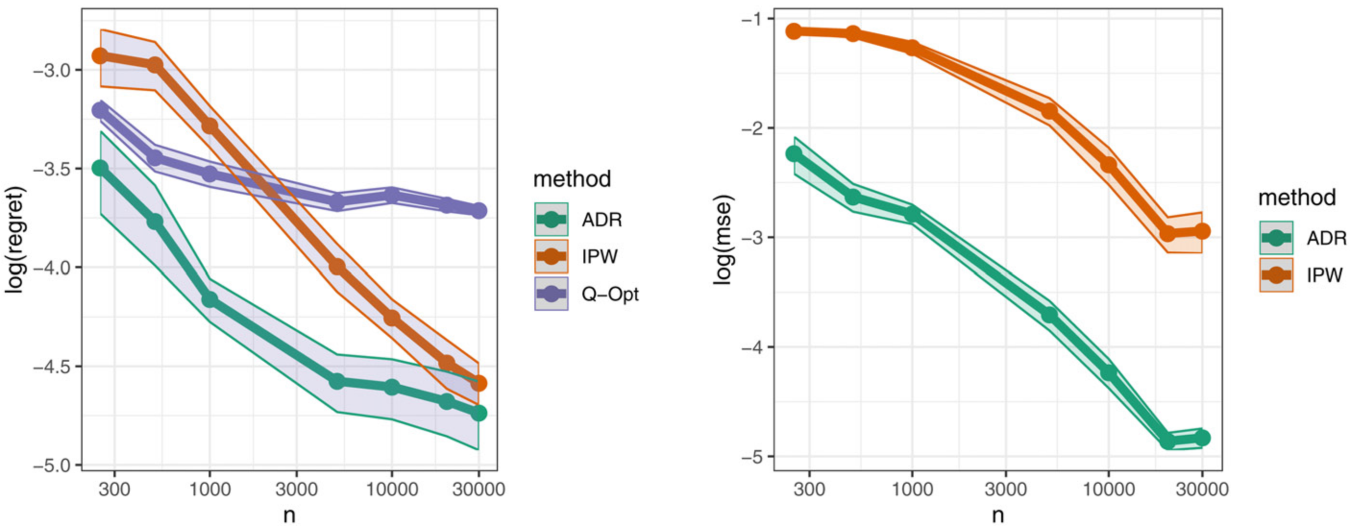

Figure 1. We compare the performance of ADR in comparison to IPW and *Q-Opt* using $\sigma = 1, \beta = 1$, and $\nu = 0.5$ in the binary treatment setup. We plot the regret (left figure) relative to the best in-class policy and the average mean-squared error (right figure) of the value estimates for policies in the same policy class across all policies (both in log-scale). The shaded regions are standard error bars. In the mean-squared error (MSE) plot, the MSE for each policy is computed against an oracle evaluation using Monte Carlo rollouts using the underlying transition dynamics averaged across 20,000 runs. Both the regret and MSE results are averaged across 50 runs. The *x*-axis shows the number of offline trajectories we generate in the observational data.

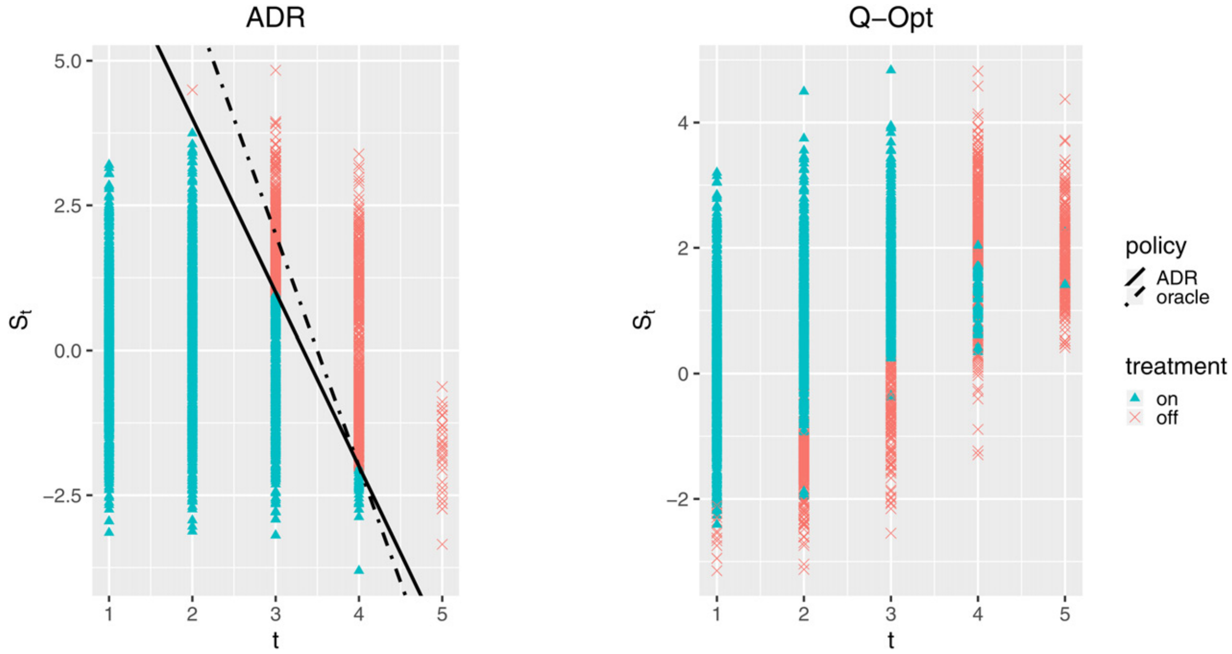

Figure 2. A single realization of the best policy learned in the binary action setup case as described in Section 5.1, in the setting of $\nu = 0.5, \beta = 5, \sigma = 1$. ADR and the oracle choose the best in-class policy from the predefined linear policy class, whereas *Q-Opt* learns the value function via blackbox regression methods and learns a policy that is not so easy to interpret. At each time step, we plot the value of the state S_t from trajectories that have not stopped treating yet.

at each time step, we show the value of the state S_t for any trajectory that has not stopped treatment yet according to the learned policy; the color coding specifies the policy decision for each trajectory at the given time step.[16] As expected, ADR learns a linear decision rule, that is, it stops treatment whenever it crosses the solid black line that parametrizes the policy (for reference, we also plot the optimal linear decision rule as a dash-dotted line). In contrast, *Q-Opt* tries to learn the optimal (unconstrained) policy, but appears to be quite noisy given the sample sizes considered here.[17]

[16] There are fewer trajectories plotted as we move along the time axis, because once a trajectory has stopped treatment, it would always stop treatment and there will be no longer decisions made.

[17] As discussed in Section 2.4, one other possible baseline—not considered here—is to use Q-learning for policy search, that is, to use *Q-Eval* to evaluate each policy, and take an argmax over the value estimate for learning. This would in general enable consistent learning over a structured policy class; however, this approach is fairly demanding computationally, as it involves separately estimating policy value for each policy via dynamic programming. One might also ask whether we could make *Q-Opt* stable and/or interpretable by using linear regression in the recursive step (42). Doing so, however, would void any nonparametric consistency guaranteed for *Q-Opt*, and in particular would not recover best-in-class linear policies. The problem is that *Q-Opt* conflates modeling and policy optimization, rather than separating out these two steps like ADR; in contrast, we first model

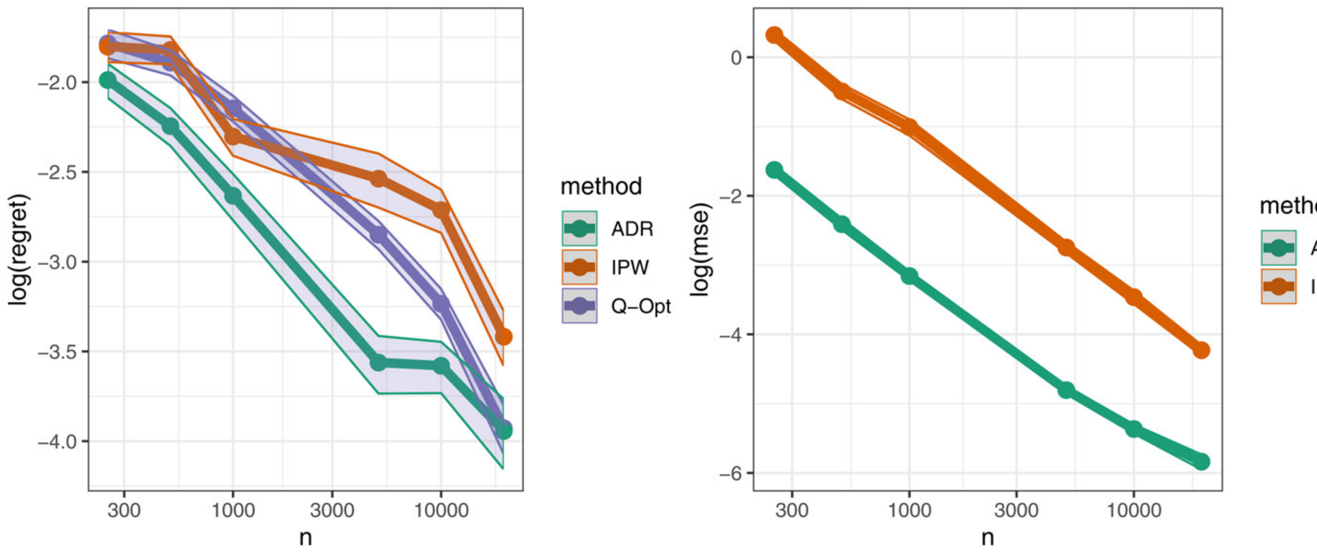

Figure 3. We compare the performance of ADR in comparison to IPW and *Q-Opt* in the multiple treatment setup. The plot shows results for $\sigma = 1$. We plot the regret (left figure) relative to the best in-class policy and the average mean-squared error (right figure) of the value estimates for policies in the same policy class across all policies (both in log-scale). The shaded regions In the mean-squared error (MSE) plot, the MSE for each policy is computed against an oracle evaluation using Monte Carlo rollouts under the underlying transition dynamics averaged across 20,000 runs. Both the regret and MSE results are averaged across 50 runs. The *x*-axis shows the number of offline trajectories we generate in the observational data.

Table 3. Detailed numerical results in the multiple-action setup.

n	σ	ADR	IPW	Q-Opt	Oracle	MSE:ADR	MSE:IPW
250	0	**1.65e-01**	1.40e-01	5.88e-02	2.65e-01	**1.79e-01**	1.35e+00
250	0.5	**1.24e-01**	9.13e-02	5.85e-02	2.67e-01	**1.83e-01**	1.26e+00
250	1	**1.17e-01**	8.94e-02	8.67e-02	2.54e-01	**1.97e-01**	1.38e+00
500	0	**1.88e-01**	1.14e-01	1.08e-01	2.65e-01	**8.74e-02**	6.39e-01
500	0.5	**1.72e-01**	1.13e-01	1.00e-01	2.67e-01	**8.30e-02**	6.55e-01
500	1	**1.48e-01**	9.30e-02	1.04e-01	2.54e-01	**9.00e-02**	6.12e-01
1000	0	**1.98e-01**	1.52e-01	1.38e-01	2.65e-01	**3.55e-02**	2.78e-01
1000	0.5	**1.99e-01**	1.40e-01	1.15e-01	2.67e-01	**4.28e-02**	3.28e-01
1000	1	**1.83e-01**	1.54e-01	1.38e-01	2.54e-01	**4.26e-02**	3.65e-01
5000	0	**2.35e-01**	1.94e-01	2.23e-01	2.65e-01	**7.04e-03**	5.99e-02
5000	0.5	**2.21e-01**	2.09e-01	2.02e-01	2.67e-01	**7.47e-03**	6.36e-02
5000	1	**2.26e-01**	1.75e-01	1.97e-01	2.54e-01	**8.21e-03**	6.43e-02
10,000	0	**2.49e-01**	2.19e-01	2.43e-01	2.65e-01	**3.36e-03**	3.04e-02
10,000	0.5	**2.36e-01**	2.06e-01	2.22e-01	2.67e-01	**3.95e-03**	3.27e-02
10,000	1	**2.27e-01**	1.88e-01	2.15e-01	2.54e-01	**4.68e-03**	3.14e-02
20,000	0	2.47e-01	2.20e-01	**2.62e-01**	2.65e-01	**1.75e-03**	1.51e-02
20,000	0.5	**2.45e-01**	2.08e-01	2.39e-01	2.67e-01	**2.02e-03**	1.47e-02
20,000	1	**2.35e-01**	2.22e-01	**2.35e-01**	2.54e-01	**2.92e-03**	1.46e-02

NOTE: In the third to the sixth columns, we show the value of the best learned policy using ADR, IPW, and *Q-Opt* against the value of the oracle best policy in the prespecified policy class, with all value estimates evaluated using a Monte Carlo rollout with 20,000 repeats. In the right two columns, we show the mean-squared error of the value estimates averaged across all policies in the policy class. Results are averaged across 50 runs and rounded to two decimal places. Numbers listed are accurate up to the second displaying digit due to sampling errors.

5.2. Multiple Treatment Choices

In the second setup, we consider multiple treatment choices. Our design here is motivated by a healthcare setting where, once a doctor starts treatment, they can choose between a more effective but more invasive treatment with strong side effects, or a less effective but less invasive treatment. More specifically, imagine a cancer patient's state at time t is modeled by X_t, Y_t, and Z, where X_t is the general health state, Y_t is the state of a tumor, and Z is not time-dependent but models the category of the patients for which lifespan differs. In particular, if $Z = 0$, a patient always dies immediately; if $Z = 1$, a patient always survives until the end of a trial; if $Z = 2$, the patient's lifespan has a strong dependency on Y_t, which we detail below. There are two treatment choices, one noninvasive ($A_t = 1$) and one invasive ($A_t = 2$). The noninvasive option lessens the severity of the tumor, and the invasive option completely removes the tumor, but exacerbates a patient's general health conditions. The final outcome is denoted by R, which is the lifetime of a patient, and we seek a policy π that maximizes $\mathbb{E}_\pi\,[R]$. We consider horizon $T = 10$. The data is generated via the following hidden Markov process, where X_t and Y_t are unobserved:

$$X_1 \sim \mathrm{Exp}(1)$$
$$Y_1 \sim 0.5\mathrm{Exp}(3)$$
$$Z \sim \mathrm{Multinomial}(0.3,\ 0.3,\ 0.4)$$
$$L_1 = 1$$
$$Z = 1 : L_{t+1} = 0,$$

μ_{now} and μ_{next} using appropriately flexible method and then choose policy $\hat{\pi}$ in a separate optimization step where we can enforce structure.

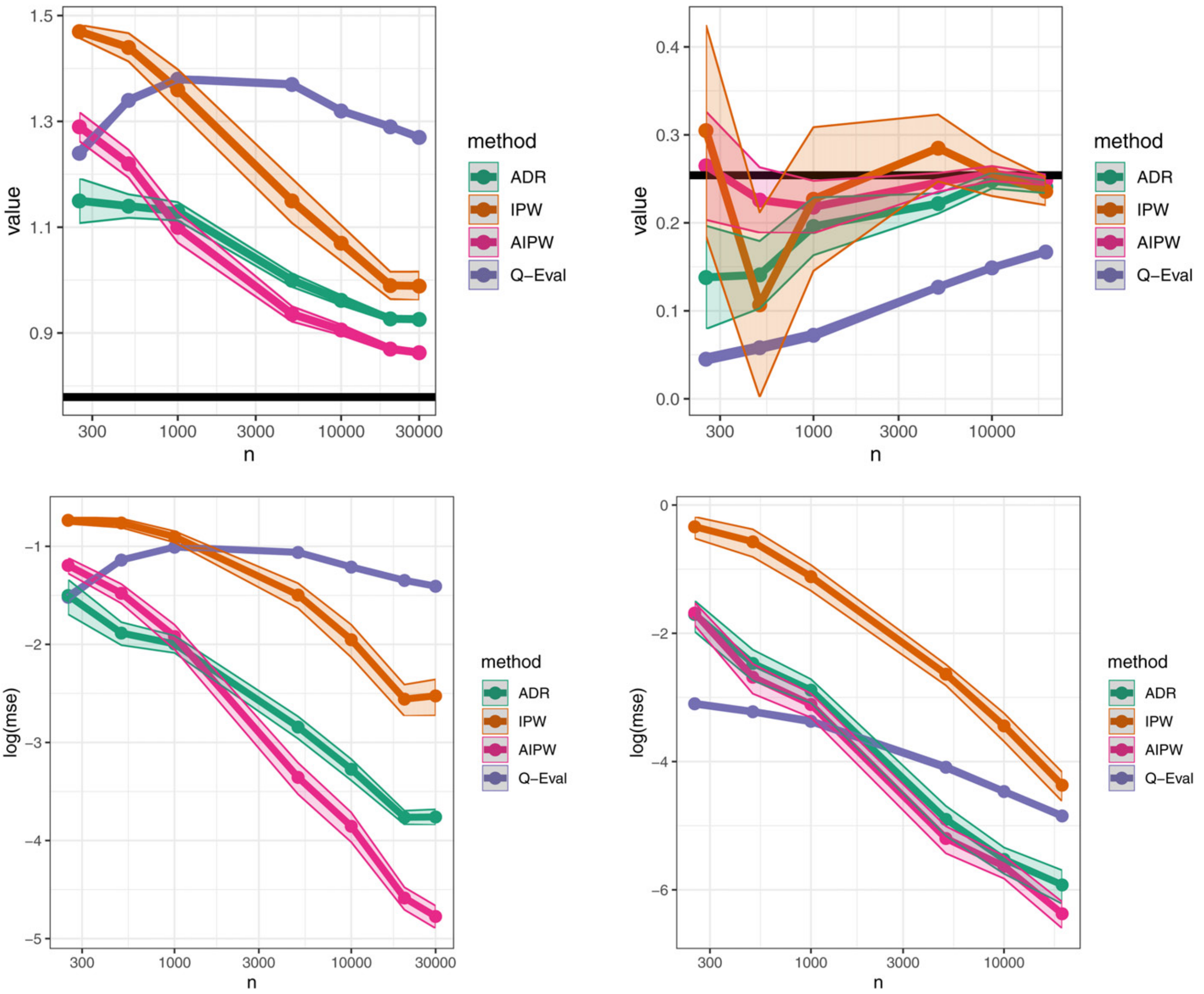

Figure 4. Comparison of ADR, IPW, AIPW, and *Q-Eval* for estimating the value improvement of the best in-class policy over the never stop policy. The left panel is in the setting of Figure 1 for the binary-treatment setup while the right panel is in the setting of Figure 3 for the multiple-treatment setup. The top two figures compare the average value estimates of the optimal policy, with the black solid line denotes the true value improvement of the optimal policy via Monte Carlo simulations over 20,000 trials. The bottom two figures are the mean squared errors (MSE) of the value estimates on learning the optimal policy. The results here are averaged across 50 independent runs, and the shaded regions denote sampling error.

$$
\begin{aligned}
Z &= 2 : L_{t+1} = 1\\
Z &= 3 : L_{t+1} = 0 \quad \text{if}\\
L_t &= 0; \qquad\qquad (43)\\
&\quad \text{otherwise},\\
L_{t+1} &\sim \text{Bernoulli}(\mathbb{1}_{Y_t \le 5} \exp(-0.02 Y_t)\\
&\qquad + \mathbb{1}_{5 < Y_t \le 14} \exp(-0.06 Y_t))\\
A_t &= 0 : X_{t+1} = |X_t + \sigma_t|\\
Y_{t+1} &= |Y_t + 0.5 X_t + \sigma_t|\\
A_t &= 1 : X_{t+1} = |X_t + \sigma_t| \qquad\qquad (44)\\
Y_{t+1} &= |0.5 Y_t + \sigma_t|\\
A_t &= 2 : X_{t+1} = X_t + \left|\max(X_t^2, 1.5 X_t) + \sigma_t - X_t\right|\\
Y_{t+1} &= 0\\
X_t' &= \max\left(0, \min\left(X_{\max}, X_t + \nu\right)\right),\\
Y_t' &= \max\left(0, \min\left(Y_{\max}, Y_t + \nu\right)\right)
\end{aligned}
$$

$$
\begin{aligned}
R &= \min\{t : L_t = 0\} - 1,\\
X_{\max} &= 10, \quad Y_{\max} = 16,\\
\sigma_t &\sim \mathcal{N}(0, 0.25), \quad \nu \sim \mathcal{N}(0, \sigma^2), \qquad (45)
\end{aligned}
$$

where L_t is an indicator for whether the patient is alive at time t.

In this setting, the treatment assignment mechanism is based on sequential randomization in the data such that there are roughly equal number of trajectories that start treating at each time with either treatment option. Note that the states we observe is X_t' and Y_t', which is the original states added with noise, making our setup non-Markovian. We consider the following linear thresholding class: $\theta_1 X_t' + \theta_2 Y_t' + \theta_3 t \ge \theta_4$ is the region in which we start treatment. If in addition, $\theta_5 X_t' + \theta_6 Y_t' + \theta_7 t \ge \theta_8$, we use the invasive treatment and otherwise, use the noninvasive treatment. We search over the eight parameters in the policy class with a grid search, with details in Appendix B in the supplementary materials.

We compare running the ADR policy optimization procedure (as shown in Section 2.3) against IPW and *Q-Opt*. Like the binary-action setup, we again estimate the oracle value of all policies in the policy class with Monte Carlo rollouts averaged across 20,000 times.

In Figure 3, we see that for both the best value learned and the average mean-squared error, ADR outperforms IPW. We also include the complete set of results with varying noise parameter σ in Table 3 in Appendix B in the supplementary materials. Interestingly, we see that in very large samples *Q-Opt* becomes competitive with ADR. One possible explanation for this is that ADR is only allowed to use linear thresholding policies whereas *Q-Opt* learns over arbitrary policies—and, in large samples, the increased expressivity of *Q-Opt* may become helpful.

5.3. Policy Learning Versus Policy Evaluation

Throughout this article, we have focused on ADR as a method for policy learning, and have emphasized that ADR is well suited to policy learning by empirical maximization because it can evaluate any policy in the policy class Π using a single set of universal scores as in (14). In contrast, standard DR methods like AIPW (7) require different nuisance components to evaluate different policies, thus making them less readily applicable to learning. That being said, it may still be of interest to compare ADR with AIPW for the task of evaluating a single policy, and to see whether the form of ADR—optimized for policy learning—sacrifices accuracy when used for evaluation.

To this end, we revisit the two simulation settings discussed above. However, instead of trying to learn the best policy, we simply seek to evaluate how much the optimal policy improves over a never-treating policy. For ADR and IPW, we use the same value estimates as were maximized for policy learning. For AIPW, we use a weighted form of (7) as in Thomas and Brunskill (2016), with value functions estimated by *Q-Eval*, that is, by a backward iteration procedure analogous to (42) that is tailored to evaluating a specific policy as opposed to finding the best policy. Finally, we also consider *Q-Eval* on its own, by averaging across the learned Q values in the initial state across the initial state distribution and the action distribution that follows the policy of interest.

Overall, as seen in Figure 4, the robust methods—ADR and AIPW—substantially outperform both IPW and *Q-Eval* here, while AIPW is slightly more accurate than ADR. It thus appears that if the only task of interest is to evaluate a prespecified policy then AIPW is a good method to start with. However, when there's also a need to choose the best among many possible policies, ADR enables us to evaluate different policies via shared outcome models and so may present a valuable option for learning via empirical maximization.

Supplementary Materials

The supplementary materials include all proofs and simulation details.

Acknowledgments

We are grateful for enlightening conversations with Susan Athey, Miguel Hernan, Michael Kosorok, Percy Liang, Susan Murphy, Jamie Robins, Andrea Rotnitzky, and Zhengyuan Zhou, as well as for helpful comments and feedback from the associate editor and seminar participants at several universities and workshops.

Funding

XN acknowledges the partial support from the Stanford Data Science Scholars program. EB acknowledges the partial support of a NSF Career Award and a Siemens grant. SW acknowledges the partial support of a Stanford Institute for Human-Centered Artificial Intelligence grant, a Facebook Faculty Award, and National Science Foundation grant DMS-1916163.

References

Antos, A., Szepesvári, C., and Munos, R. (2008a), "Fitted Q-Iteration in Continuous Action-Space MDPs," in *Advances in Neural Information Processing Systems*, pp. 9–16. [398]

——— (2008b), "Learning Near-Optimal Policies With Bellman-Residual Minimization Based Fitted Policy Iteration and a Single Sample Path," *Machine Learning*, 71, 89–129. [398]

Athey, S., and Imbens, G. (2018), "Design-Based Analysis in Difference-in-Differences Settings With Staggered Adoption," arXiv no. 1808.05293. [393]

Athey, S., Tibshirani, J., and Wager, S. (2019), "Generalized Random Forests," *The Annals of Statistics*, 47, 1148–1178. [402]

Athey, S., and Wager, S. (2020), "Policy Learning With Observational Data," *Econometrica* (forthcoming). [392,394,395,396,397,398]

Bertsimas, D., and Kallus, N. (2020), "From Predictive to Prescriptive Analytics," *Management Science*, 66, 1025–1044. [392]

Chakraborty, B., and Moodie, E. (2013), *Statistical Methods for Dynamic Treatment Regimes*, New York: Springer. [397]

Chen, J., and Jiang, N. (2019), "Information-Theoretic Considerations in Batch Reinforcement Learning," in *Proceedings of International Conference on Machine Learning*. [397]

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., and Robins, J. (2018), "Double/Debiased Machine Learning for Treatment and Structural Parameters," *The Econometrics Journal*, 21, C1–C68. [396]

Clifton, J., and Laber, E. (2020), "Q-Learning: Theory and Applications," *Annual Review of Statistics and Its Application*, 7, 279–301. [397]

Doroudi, S., Thomas, P. S., and Brunskill, E. (2017), "Importance Sampling for Fair Policy Selection," in *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*. [394,397]

Dudík, M., Erhan, D., Langford, J., and Li, L. (2014), "Doubly Robust Policy Evaluation and Optimization," *Statistical Science*, 29, 485–511. [392,394]

Elmachtoub, A. N., and Grigas, P. (2017), "Smart 'Predict, Then Optimize'," arXiv no. 1710.08005. [392]

Ernst, D., Geurts, P., and Wehenkel, L. (2005), "Tree-Based Batch Mode Reinforcement Learning," *Journal of Machine Learning Research*, 6, 503–556. [397]

Farahmand, A.-M., Ghavamzadeh, M., Szepesvári, C., and Mannor, S. (2016), "Regularized Policy Iteration With Nonparametric Function Spaces," *Journal of Machine Learning Research*, 17, 4809–4874. [397]

Goel, K., Dann, C., and Brunskill, E. (2017), "Sample Efficient Policy Search for Optimal Stopping Domains," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 1711–1717. [393,398]

Haussler, D. (1995), "Sphere Packing Numbers for Subsets of the Boolean *n*-Cube With Bounded Vapnik-Chervonenkis Dimension," *Journal of Combinatorial Theory*, Series A, 69, 217–232. [399]

Hernán, M. A., Brumback, B., and Robins, J. M. (2001), "Marginal Structural Models to Estimate the Joint Causal Effect of Nonrandomized Treatments," *Journal of the American Statistical Association*, 96, 440–448. [393]

Imbens, G. W., and Rubin, D. B. (2015), *Causal Inference in Statistics, Social, and Biomedical Sciences*, New York: Cambridge University Press. [393]

Jacka, S. D. (1991), "Optimal Stopping and the American Put," *Mathematical Finance*, 1, 1–14. [393,398]

Jiang, N., and Li, L. (2016), "Doubly Robust Off-Policy Value Evaluation for Reinforcement Learning," in *Proceedings of the 33rd International Conference on Machine Learning*, Proceedings of Machine Learning Research (PMLR). [392,394,396,397]

Kakade, S. M. (2003), "On the Sample Complexity of Reinforcement Learning," PhD thesis, University of London, London, England. [395]

Kallus, N. (2018), "Balanced Policy Evaluation and Learning," in *Advances in Neural Information Processing Systems*, pp. 8909–8920. [394]

Kallus, N., and Uehara, M. (2020), "Statistically Efficient Off-Policy Policy Gradients," in *International Conference on Machine Learning*. [398]

Kallus, N., and Zhou, A. (2018), "Confounding-Robust Policy Improvement," in *Advances in Neural Information Processing Systems*, pp. 9269–9279. [392]

Kitagawa, T., and Tetenov, A. (2018), "Who Should Be Treated? Empirical Welfare Maximization Methods for Treatment Choice," *Econometrica*, 86, 591–616. [392,394,397,398]

Kosorok, M. R., and Laber, E. B. (2019), "Precision Medicine," *Annual Review of Statistics and Its Application*, 6, 263–286. [397]

Lavori, P. W., and Dawson, R. (2000), "A Design for Testing Clinical Strategies: Biased Adaptive Within-Subject Randomization," *Journal of the Royal Statistical Society*, Series A, 163, 29–38. [397]

Le, H., Voloshin, C., and Yue, Y. (2019), "Batch Policy Learning Under Constraints," in *International Conference on Machine Learning*, pp. 3703–3712. [397,402]

Levine, S., Kumar, A., Tucker, G., and Fu, J. (2020), "Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems," arXiv no. 2005.01643. [397]

Liu, Q., Li, L., Tang, Z., and Zhou, D. (2018), "Breaking the Curse of Horizon: Infinite-Horizon Off-Policy Estimation," in *Advances in Neural Information Processing Systems*, pp. 5361–5371. [398]

Liu, Y., Gottesman, O., Raghu, A., Komorowski, M., Faisal, A. A., Doshi-Velez, F., and Brunskill, E. (2018), "Representation Balancing MDPs for Off-Policy Policy Evaluation," in *Advances in Neural Information Processing Systems*, pp. 2649–2658. [394]

Liu, Y., Swaminathan, A., Agarwal, A., and Brunskill, E. (2019), "Off-Policy Policy Gradient With State Distribution Correction," in *Proceedings of Uncertainty in AI*. [397]

Luckett, D. J., Laber, E. B., Kahkoska, A. R., Maahs, D. M., Mayer-Davis, E., and Kosorok, M. R. (2020), "Estimating Dynamic Treatment Regimes in Mobile Health Using V-Learning," *Journal of the American Statistical Association*, 115, 692–706. [392,398]

Luedtke, A., and Chambaz, A. (2017), "Faster Rates for Policy Learning," arXiv no. 1704.06431. [399]

Manski, C. F. (2004), "Statistical Treatment Rules for Heterogeneous Populations," *Econometrica*, 72, 1221–1246. [392,394,397,398]

Moodie, E. E., Platt, R. W., and Kramer, M. S. (2009), "Estimating Response-Maximized Decision Rules With Applications to Breastfeeding," *Journal of the American Statistical Association*, 104, 155–165. [392]

Mordecki, E. (2002), "Optimal Stopping and Perpetual Options for Lévy Processes," *Finance and Stochastics*, 6, 473–493. [398]

Munos, R. (2003), "Error Bounds for Approximate Policy Iteration," in *ICML* (Vol. 3), pp. 560–567. [397]

Munos, R., and Szepesvári, C. (2008), "Finite-Time Bounds for Fitted Value Iteration," *Journal of Machine Learning Research*, 9, 815–857. [397,398]

Murphy, S. A. (2003), "Optimal Dynamic Treatment Regimes," *Journal of the Royal Statistical Society*, Series B, 65, 331–355. [392,393,397]

——— (2005), "A Generalization Error for Q-Learning," *Journal of Machine Learning Research*, 6, 1073–1097. [392,393,395,397,402]

Neyman, J. (1923), "Sur les applications de la théorie des probabilités aux experiences agricoles: Essai des principes," *Roczniki Nauk Rolniczych*, 10, 1–51. [393]

Prasad, N., Cheng, L.-F., Chivers, C., Draugelis, M., and Engelhardt, B. E. (2017), "A Reinforcement Learning Approach to Weaning of Mechanical Ventilation in Intensive Care Units," in *Conference on Uncertainty in Artificial Intelligence*. [392]

Precup, D. (2000), "Eligibility Traces for Off-Policy Policy Evaluation," Computer Science Department Faculty Publication Series, p. 80. [394]

Robins, J. (1986), "A New Approach to Causal Inference in Mortality Studies With a Sustained Exposure Period: Application to Control of the Healthy Worker Survivor Effect," *Mathematical Modelling*, 7, 1393–1512. [393,397]

——— (2004), "Optimal Structural Nested Models for Optimal Sequential Decisions," in *Proceedings of the second Seattle Symposium in Biostatistics*, Springer, pp. 189–326. [392,393,397]

Robins, J. M., Hernán, M. A., and Brumback, B. (2000), "Marginal Structural Models and Causal Inference in Epidemiology," *Epidemiology*, 11, 551, 2000. [394]

Robins, J. M., and Rotnitzky, A. (1995), "Semiparametric Efficiency in Multivariate Regression Models With Missing Data," *Journal of the American Statistical Association*, 90, 122–129. [396]

Robins, J. M., Rotnitzky, A., and Zhao, L. P. (1994), "Estimation of Regression Coefficients When Some Regressors Are Not Always Observed," *Journal of the American Statistical Association*, 89, 846–866. [395]

Rosenbaum, P. R., and Rubin, D. B. (1983), "The Central Role of the Propensity Score in Observational Studies for Causal Effects," *Biometrika*, 70, 41–55. [393]

Rubin, D. B. (1974), "Estimating Causal Effects of Treatments in Randomized and Nonrandomized Studies," *Journal of Educational Psychology*, 66, 688. [393]

Rust, J. (1987), "Optimal Replacement of GMC Bus Engines: An Empirical Model of Harold Zurcher," *Econometrica: Journal of the Econometric Society*, 55, 999–1033. [398]

Schick, A. (1986), "On Asymptotically Efficient Estimation in Semiparametric Models," *The Annals of Statistics*, 14, 1139–1151. [396]

Sutton, R. S., and Barto, A. G. (2018), *Reinforcement Learning: An Introduction*, Cambridge, MA: MIT Press. [393,397]

Swaminathan, A., and Joachims, T. (2015), "Batch Learning From Logged Bandit Feedback Through Counterfactual Risk Minimization," *Journal of Machine Learning Research*, 16, 1731–1755. [392,394,397,398]

Thomas, P., and Brunskill, E. (2016), "Data-Efficient Off-Policy Policy Evaluation for Reinforcement Learning," in *International Conference on Machine Learning*, pp. 2139–2148. [392,394,396,397,407]

Tsiatis, A. A., Davidian, M., Holloway, S. T., and Laber, E. B. (2019), *Dynamic Treatment Regimes: Statistical Methods for Precision Medicine*, Boca Raton, FL: CRC Press. [392,395,397]

Uehara, M., and Jiang, N. (2019), "Minimax Weight and Q-Function Learning for Off-Policy Evaluation," arXiv no. 1910.12809. [398]

Van der Laan, M. J., and Rose, S. (2018), *Targeted Learning in Data Science*, Cham: Springer. [392,395]

Van Moerbeke, P. (1976), "On Optimal Stopping and Free Boundary Problems," *Archive for Rational Mechanics and Analysis*, 60, 101–148. [393,398]

Vansteelandt, S., and Joffe, M. (2014), "Structural Nested Models and G-Estimation: The Partially Realized Promise," *Statistical Science*, 29, 707–731. [397]

Watkins, C. J., and Dayan, P. (1992), "Q-Learning," *Machine Learning*, 8, 279–292. [397]

When To Start Consortium (2009), "Timing of Initiation of Antiretroviral Therapy in AIDS-Free HIV-1-Infected Patients: A Collaborative Analysis of 18 HIV Cohort Studies," *The Lancet*, 373, 1352–1363. [392]

Zanette, A., and Brunskill, E. (2019), "Tighter Problem-Dependent Regret Bounds in Reinforcement Learning Without Domain Knowledge Using Value Function Bounds," in *International Conference on Machine Learning*, pp. 7304–7312. [399]

Zhang, B., Tsiatis, A. A., Davidian, M., Zhang, M., and Laber, E. (2012), "Estimating Optimal Treatment Regimes From a Classification Perspective," *Stat*, 1, 103–114. [392,394]

Zhang, B., Tsiatis, A. A., Laber, E. B., and Davidian, M. (2013), "Robust Estimation of Optimal Dynamic Treatment Regimes for Sequential Treatment Decisions," *Biometrika*, 100, 681–694. [392,394,395,396,397]

Zhang, Y., Laber, E. B., Davidian, M., and Tsiatis, A. A. (2018), "Interpretable Dynamic Treatment Regimes," *Journal of the American Statistical Association*, 113, 1541–1549. [392,397]

Zhang, Y., Laber, E. B., Tsiatis, A., and Davidian, M. (2015), "Using Decision Lists to Construct Interpretable and Parsimonious Treatment Regimes," *Biometrics*, 71, 895–904. [395]

Zhao, Y.-Q., Zeng, D., Laber, E. B., and Kosorok, M. R. (2015), "New Statistical Learning Methods for Estimating Optimal Dynamic Treatment Regimes," *Journal of the American Statistical Association*, 110, 583–598. [394,397]

Zhao, Y., Zeng, D., Rush, A. J., and Kosorok, M. R. (2012), "Estimating Individualized Treatment Rules Using Outcome Weighted Learning," *Journal of the American Statistical Association*, 107, 1106–1118. [392,394,397]

Zhou, X., Mayer-Hamblett, N., Khan, U., and Kosorok, M. R. (2017), "Residual Weighted Learning for Estimating Individualized Treatment Rules," *Journal of the American Statistical Association*, 112, 169–187. [394]

Zhou, Z., Athey, S., and Wager, S. (2018), "Offline Multi-Action Policy Learning: Generalization and Optimization," arXiv no. 1810.04778. [392,399]