
Interpretable Off-Policy Learning via Hyperbox Search

Daniel Tschernutter¹ Tobias Hatt¹ Stefan Feuerriegel^{1,2}

Abstract

Personalized treatment decisions have become an integral part of modern medicine. Thereby, the aim is to make treatment decisions based on individual patient characteristics. Numerous methods have been developed for learning such policies from observational data that achieve the best outcome across a certain policy class. Yet these methods are rarely interpretable. However, interpretability is often a prerequisite for policy learning in clinical practice. In this paper, we propose an algorithm for *interpretable* off-policy learning via hyperbox search. In particular, our policies can be represented in disjunctive normal form (i. e., OR-of-ANDs) and are thus intelligible. We prove a universal approximation theorem that shows that our policy class is flexible enough to approximate any measurable function arbitrarily well. For optimization, we develop a tailored column generation procedure within a branch-and-bound framework. Using a simulation study, we demonstrate that our algorithm outperforms state-of-the-art methods from *interpretable* off-policy learning in terms of regret. Using real-world clinical data, we perform a user study with actual clinical experts, who rate our policies as highly interpretable.

1. Introduction

Personalized treatment decisions have received wide attention in clinical practice (Chan & Ginsburg, 2011; Collins & Varmus, 2015; Hamburg & Collins, 2010). Thereby, the aim is to select treatments that are effective for individual patients. One way to formalize personalized decision-making are so-called policies, which map patient-level characteristics to treatment decisions. Numerous methods have been developed for learning such policies using data collected

from existing clinical trials or observational studies, which are subsumed under *off-policy learning* (Atthey & Wager, 2021; Beygelzimer & Langford, 2009; Dudík et al., 2014; Kallus, 2018; Kallus & Zhou, 2018; Kitagawa & Tetenov, 2018).

A major challenge for off-policy learning in clinical practice is to learn policies that satisfy two requirements: (i) the policy class must be *sufficiently rich* and (ii) the policies must be *interpretable* (Kattan et al., 2016; Kosorok & Laber, 2019; Blümlein et al., 2022). A policy class is said to be *sufficiently rich* if it contains a large class of policies. This is desirable as it is directly related to policies that yield a low regret (i. e., the difference between clinical outcome across the population of the learned policy and the a priori best policy should be small). A policy is said to be *interpretable* if its decision can be explained in understandable terms to humans (Doshi-Velez & Kim, 2017). Particularly, in a clinical setting, it is crucial for clinical practitioners to understand *which* treatment is chosen *when* (Ahmad et al., 2018).

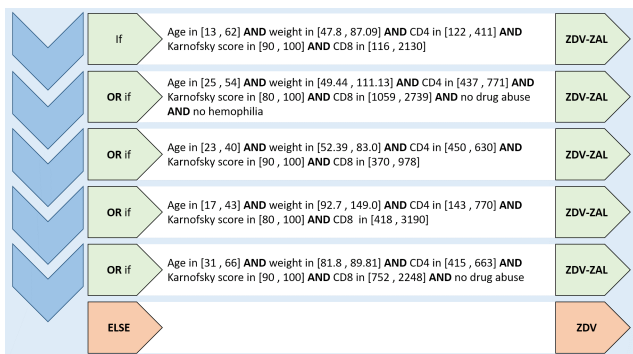
Logical models can achieve interpretability by providing a few descriptive rules that can help clinical practitioners to understand the decision process and get an intuition about the underlying data (Wang et al., 2017; Rudin, 2019). Thereby, each rule is a conjunction of conditions (Wang et al., 2017). As an example, a set of rules for deciding whether or not a patient should receive a specific treatment with a drug could be:

IF a patient	(has an age ≤ 60 AND a weight ≤ 80 kg AND has mild symptoms)
OR	(has an age ≤ 40 AND a weight ≤ 80 kg AND has severe symptoms)
OR	(has an age ≤ 30 AND a weight ≤ 100 kg AND has severe symptoms)
THEN	treat with drug

The above model suggests treatments and provides patient characteristics that lead to the decision. As such, it makes it easy for domain experts to understand and critically assess the underlying logic. For instance, in the above example, one observes that younger patients seem to better respond to the drug treatment even if they have severe symptoms and/or are too heavy. Thus, the age of a patient seems to be crucial for when it is recommended to apply the drug. Furthermore, patients over 60 years should not receive the

¹ETH Zurich, Switzerland ²LMU, Germany. Correspondence to: Daniel Tschernutter <dtchernutter@ethz.ch>, Tobias Hatt <thatt@ethz.ch>.

Figure 1. Illustration of our interpretable policies.



Notes. The policy was estimated using real-world clinical data from the ACTG study 175; see Section 5.2 for details. A clinical practitioner follows the flow chart from top to bottom (as indicated by the blue arrows). If the patient fulfills an if-clause, a practitioner should assign the treatment in the corresponding arrow to the right. We provide details on the two treatments (ZDV-ZAL and ZDV) in Section 5.2.

drug treatment in general. Based on the decision logic, clinical practitioners can compare the suggested treatment rule against their domain knowledge, e. g., they may know that the molecular of the drug is non-functioning for older patients or has severe side-effects for them.

Besides being interpretable, a crucial part of logical models is the form of the conditions (Rudin, 2019), e. g., age ≤ 60 in the above example. The reason is that, even in simple settings, the best policy is often a nonlinear and complex function of patient characteristics (Laber et al., 2014; Robins, 2004; Schulte et al., 2014). Hence, flexible policy classes should be used in order to mitigate the risk of model misspecification. For logical models, it is generally known that such flexibility results in computationally hard problems (Rudin, 2019). Thus, the challenge is to formulate expressive conditions in a way that allows for efficient optimization.

In this paper, we address the problem of learning *interpretable* policies from observational data, where the policies should originate from a sufficiently rich policy class. To this end, we propose an algorithm for *interpretable off-policy learning* called IOPL. IOPL relies on an intelligible yet flexible class of policies. Specifically, our policy class is based on a representation of the decision region as a union of hyperboxes. As a result, our policies can be written in disjunctive normal form (i. e., OR-of-ANDs) and, thus, belong to the class of logical models. In particular, this representation gives rules which are conjunctions of interval conditions, e. g., age $\in [20, 60]$. These are easy enough to be intelligible but expressive enough to approximate any measurable function arbitrarily well. An example of such a policy is shown in Figure 1.

Optimizing over this policy class, i. e., learning optimal

unions of hyperboxes, relies on a mixed integer linear program formulation. The latter involves an exponentially growing number of variables, which renders the optimization problem non-trivial. As a remedy, we develop a tailored optimization algorithm in form of a column generation procedure within a branch-and-bound framework (also known as branch-and-price). In this way, IOPL is able to efficiently search over the exponential number of hyperboxes and yields optimal solutions with theoretical guarantees. In addition, we prove a universal approximation theorem for our policy class: we show that our policies are flexible enough to approximate any measurable policy arbitrarily well. As a result, our policy class is sufficiently expressive to learn policies that are complex functions of patient characteristics. Using synthetic data, we demonstrate that IOPL outperforms existing baselines for interpretable off-policy learning in terms of regret. Furthermore, the regret of IOPL is on par with “black box” approaches (that are more flexible yet non-intelligible). We further confirm that IOPL yields policies that are perceived as interpretable in clinical practice. For this, we use real clinical data and perform a user study in which clinical experts assess the interpretability of our policies.

2. Problem Setup

We consider n independent and identically distributed samples $\{(X_i, T_i, Y_i)\}_{i=1}^n$, where $X \in \mathcal{X} \subseteq \mathbb{R}^d$ denotes the patient covariates, $T \in \{-1, 1\}$ denotes the treatment assignment, and $Y \in \mathbb{R}$ denotes the observed outcome. We use the convention that lower outcomes are more desirable. Using the Rubin-Neyman potential outcomes framework (Rubin, 2005), we let $Y(1), Y(-1)$ be the potential outcomes for each of the treatments. To ensure identifiability of the outcomes from observational data, we make the following three standard assumptions (Imbens & Rubin, 2015): (i) consistency (i. e., $Y = Y(T)$); (ii) positivity (i. e., $0 < \mathbb{P}(T = t | X = x)$ for $t \in \{-1, 1\}$ and all x); and (iii) strong ignorability (i. e., $Y(-1), Y(1) \perp\!\!\!\perp T | X$).

We let a *policy* π be a map from the patient covariates to a treatment decision, i. e., $\pi : \mathcal{X} \rightarrow \{-1, 1\}$. Then, the *policy value* of π is given by

$$V(\pi) = \mathbb{E}[Y(\pi(X))] = \mathbb{E}[Y(1)I(\pi(X) = 1) + Y(-1)I(\pi(X) = -1)], \quad (1)$$

which gives the population-wide outcome if treatments are assigned according to the policy π . The objective of *policy learning* is to find a policy π^{opt} in a policy class Π that minimizes the policy value $V(\pi)$, i. e.,

$$\pi^{\text{opt}} \in \arg \min_{\pi \in \Pi} V(\pi). \quad (2)$$

Most existing methods first estimate the policy value $V(\pi)$ in (1) and then optimize over a pre-specified policy class

II. The policy class Π is often chosen such that π^{opt} achieves a low regret, i. e., such that $\text{Regret}(\pi^{\text{opt}}) = \min_{\pi \in \{-1, 1\}^{\mathcal{X}}} V(\pi) - V(\pi^{\text{opt}})$ is small.¹ Hence, the policy class Π must be rich enough to capture complex dependencies between patient covariates and outcome. This is usually achieved by choosing flexible policy classes such as kernel methods or neural networks (Bennett & Kallus, 2020; Laber et al., 2014; Qian & Murphy, 2011; Zhao et al., 2019; Zhou et al., 2017). Even though flexible policy classes mitigate the risk of model misspecification, they render the estimated policy unintelligible. That is, interpretability is precluded.

3. Related Work

In this section, we briefly review the literature on off-policy learning. As IOPL relies on the solution of a large-scale combinatorial optimization problem, we further discuss a technique called branch-and-price to solve such problems. The latter represents the foundation of IOPL as discussed in more detail later on.

Off-policy learning. There is a large literature on learning policies using data from observational data. Methods have been developed in research from both statistics (Blatt et al., 2004; Murphy, 2003; Pan & Zhao, 2021; Qian & Murphy, 2011; Robins, 2004; Zhao et al., 2019; Zhou et al., 2017) and machine learning (Athey et al., 2017; Athey & Wager, 2021; Beygelzimer & Langford, 2009; Dudík et al., 2014; Kallus, 2018; Kallus & Zhou, 2018; Kitagawa & Tetenov, 2018; Zhou et al., 2018). Therein, the authors first estimate the policy value in (1) and then optimize the policy value over a pre-specified policy class Π . Estimators for the policy value can be broadly divided into three categories: (i) *Direct methods* (DM) estimate the outcome functions $\mu_t(x) = \mathbb{E}[Y(t) | X = x]$ and plug them into Equation (1) (Bennett & Kallus, 2020; Qian & Murphy, 2011). Direct methods are known to be weak against model misspecification with regards to $\mu_t(x)$. (ii) *Inverse propensity score weighted methods* (IPS) seek weights that render the outcome data as if it were generated by a new policy (Beygelzimer & Langford, 2009; Bottou et al., 2013; Horvitz & Thompson, 1952; Kallus, 2018; Kallus & Zhou, 2018; Li et al., 2011). As such, IPS methods correct for the distributional mismatch between the outcomes observed under the behavioral policy and the outcomes that would be observed under a new policy. (iii) *Doubly robust methods* (DR) combine direct and IPS methods, typically using the augmented inverse propensity score estimator (Thomas & Brunskill, 2016; Dudík et al., 2014; Athey & Wager, 2021). As a result, either the direct method or the IPS method need to be consistent for the doubly robust estimator to be consistent (Dudík et al., 2014; Thomas & Brunskill, 2016). Once an

¹Thereby, we used $\{-1, 1\}^{\mathcal{X}}$ to denote all functions from \mathcal{X} to $\{-1, 1\}$, i. e., all $\pi : \mathcal{X} \rightarrow \{-1, 1\}$.

estimate of the policy value is obtained, the authors optimize over a pre-specified Π . In order to allow for complex policies, flexible policy classes such as kernel methods or neural networks are used (Bennett & Kallus, 2020; Laber et al., 2014; Qian & Murphy, 2011; Zhao et al., 2019; Zhou et al., 2017). Yet, the interpretability of such policies is precluded, which limits the use in clinical practice (Rudin, 2019).

Approaches for interpretable off-policy learning include works based on decision lists, i. e., lists of “if-then” clauses (Lakkaraju & Rudin, 2017; Zhang et al., 2018; 2015), eligibility scores (Kitagawa & Tetenov, 2018) or decision trees (Laber & Zhao, 2015). However, besides being interpretable, none of these provide any theoretical guarantees that the used policy class is sufficiently rich to capture complex policies as we do.

Branch-and-price. Branch-and-price is a general method to solve large scale integer linear programs. For a detailed introduction, we refer to (Barnhart et al., 1998). Branch-and-price is based on the idea to use column generation within an enumeration framework, i. e., within a branch-and-bound approach. Column generation itself has already been used in machine learning research (Bi et al., 2004; Demiriz et al., 2002; Li et al., 2013). Furthermore, column generation has also been used to derive interpretable classifiers (Dash et al., 2018) but not in the context of off-policy learning, resulting in three major differences to our work: (i) Dash et al. consider general binary classification problems using the hamming loss, while we minimize the policy value over a certain policy class. (ii) The authors merely develop a column generation framework for their relaxed problem formulation. The authors do not provide a branch-and-price approach as we do. Thus, they also offer no general convergence guarantees. (iii) The authors make the assumption that only binary-valued features are given, whereas we refrain from any structural assumptions.

4. Interpretable Off-Policy Learning

In this section, we introduce our algorithm for interpretable off-policy learning. We proceed as follows. In Section 4.1, we introduce a policy class Π_H^M based on hyperboxes that is sufficiently rich while, at the same time, offers an intelligible structure and thus allows for interpretability. In Section 4.2, we reformulate the empirical version of (2) as a mixed integer linear program. The latter is intractable due to an exponentially growing number of variables and is only for theoretical purposes to explain our conceptual framework. In Section 4.3, we yield theoretical results regarding the approximation capability of our policy class Π_H^M , while in Section 4.5, we introduce IOPL, which uses a tailored branch-and-price procedure to solve the aforementioned mixed integer linear program.

4.1. Interpretable Policy Class Π_H^M via Hyperboxes

In the following, we formalize our policy class Π_H^M consisting of policies that are built upon unions of hyperboxes. To do so, let $\mathcal{K} \subseteq 2^{\mathbb{R}^d}$ denote a given finite set of hyperboxes in \mathbb{R}^d , i.e., $\mathcal{K} = \{S^1, \dots, S^N\}$ with $S^j = [l_1^j, u_1^j] \times [l_2^j, u_2^j] \times \dots \times [l_d^j, u_d^j]$ for $j \in \{1, \dots, N\}$. We consider policies of the form

$$\pi_{\mathcal{D}}(X) = \begin{cases} 1, & \text{if } \exists j \in \mathcal{D} \subseteq \{1, \dots, N\} : X \in S^j, \\ -1, & \text{else,} \end{cases} \quad (3)$$

where \mathcal{D} denotes which of the N hyperboxes are used for the treatment decision. Furthermore, we bound the number of used hyperboxes by M . That is, we define the policy class $\Pi_H^M = \{\pi_{\mathcal{D}} : \mathcal{D} \subseteq \{1, \dots, N\} \text{ and } |\mathcal{D}| \leq M\}$. The policies in Π_H^M can be represented as OR-of-ANDs and are thus in disjunctive normal form, which is deemed “interpretable” in the literature (Rudin, 2019). An example is given in Figure 1. We offer an in-depth discussion on the interpretability of our approach in Appendix A.

4.2. Mixed Integer Linear Program Formulation

Optimizing the policy value over Π_H^M as in (2) is non-trivial. In the off-policy learning literature, (2) is often reformulated as a binary classification problem. Our work also builds upon such a formulation, as it allows to reformulate the empirical version as a mixed integer linear program, which, in turn, allows for a column generation approach when relaxed (we show this in Sec. 4.5). This will be the key ingredient of IOPL.

We proceed as follows. Problem (2) is equivalent to

$$\pi^{\text{opt}} \in \arg \min_{\pi \in \Pi} \mathcal{J}(\pi) \quad (4)$$

with $\mathcal{J}(\pi) = \mathbb{E}[\psi I(T \neq \pi(X))]$, where I is the indicator function and ψ corresponds to one of the three standard methods for policy learning: direct method ψ^{DM} , inverse propensity score weighted method ψ^{IPS} , and doubly robust method ψ^{DR} . These are defined as

$$\psi^{\text{DM}} = T(\mu_{-1}(X) - \mu_1(X)), \quad (5)$$

$$\psi^{\text{IPS}} = \frac{-Y}{e_T(X)}, \quad (6)$$

$$\psi^{\text{DR}} = \psi^{\text{DM}} + \psi^{\text{IPS}} + \frac{\mu_T(X)}{e_T(X)}. \quad (7)$$

The nuisance functions $\mu_t(x) = \mathbb{E}[Y(t) | X = x]$ and $e_t(x) = \mathbb{P}(T = t | X = x)$ can be estimated from data. For details on this reformulation, see Appendix B. Then, we approximate (4) with its empirical version resulting in the empirical binary classification problem

$$\min_{\pi \in \Pi} \mathcal{J}_n(\pi), \quad (\text{EBCP})$$

where $\mathcal{J}_n(\pi) = \frac{1}{n} \sum_{i=1}^n \psi_i I(T_i \neq \pi(X_i))$. By setting $\Pi = \Pi_H^M$ in (EBCP), it is sufficient to consider only hyperboxes spanned by the subsets of $\{X_i\}_{i=1}^n$, i.e., hyperboxes with $l_t = \min_{i \in \mathcal{I}} X_{i,t}$ and $u_t = \max_{i \in \mathcal{I}} X_{i,t}$ for each $t \in \{1, \dots, d\}$, where $\mathcal{I} \subseteq \{1, \dots, n\}$ and $X_{i,t}$ denotes the t -th coordinate of X_i .² That is, the set \mathcal{K} is indeed finite. However, this results in at most 2^n different boxes, i.e., $N \approx 2^n$, which calls for an efficient optimization approach. For this, (EBCP) can be reformulated as a mixed integer linear program as follows. From now on, we assume w.l.o.g. that $\psi_i \neq 0$ for all $i \in \{1, \dots, n\}$. We split the set $\{1, \dots, n\}$ into indices containing positive values for ψ and indices containing negative ones, i.e., $\mathcal{P} = \{i \in \{1, \dots, n\} : \psi_i > 0\}$ and $\mathcal{N} = \{i \in \{1, \dots, n\} : \psi_i < 0\}$. Furthermore, let $I_t = \{i \in \{1, \dots, n\} : T_i = t\}$ for $t \in \{-1, 1\}$ denote the index sets of treated and untreated individuals and let \mathcal{K}_i denote the set of all indices $j \in \{1, \dots, N\}$ such that $X_i \in S^j$. We introduce decision variables $s_j \in \{0, 1\}$, which denote whether hyperbox S^j is used in the policy or not, i.e., $\mathcal{D} = \{j : s_j = 1\}$. Furthermore, let ξ_i denote whether individual i was treated according to $\pi_{\mathcal{D}}$ or not, i.e., $\xi_i = I(T_i \neq \pi_{\mathcal{D}}(X_i))$. The mixed integer linear program formulation of (EBCP), which we denote by (MILP), is then as follows:

$$\min_{s, \xi} \frac{1}{n} \sum_{i=1}^n \psi_i \xi_i \quad (8)$$

$$\text{s.t. } \xi_i + \sum_{j \in \mathcal{K}_i} s_j \geq 1 \quad \text{for } i \in I_1 \cap \mathcal{P}, \quad (9)$$

$$\xi_i \geq s_j \quad \text{for } i \in I_{-1} \cap \mathcal{P} \text{ and } j \in \mathcal{K}_i, \quad (10)$$

$$\xi_i \leq 1 - s_j \quad \text{for } i \in I_1 \cap \mathcal{N} \text{ and } j \in \mathcal{K}_i, \quad (11)$$

$$\xi_i \leq \sum_{j \in \mathcal{K}_i} s_j \quad \text{for } i \in I_{-1} \cap \mathcal{N}, \quad (12)$$

$$\sum_{j=1}^N s_j \leq M, \quad (13)$$

$$s_j \in \{0, 1\}, \xi_i \in [0, 1]. \quad (14)$$

Note that (MILP) is equivalent to (EBCP) for $\Pi = \Pi_H^M$. See Appendix C for a detailed discussion on the equivalence. Constraints (9) and (11) enforce the equality $\xi_i = I(T_i \neq \pi_{\mathcal{D}}(X_i))$ for $i \in I_1$, while constraints (10) and (12) enforce the same equality for $i \in I_{-1}$. Constraint (13) introduces the upper bound on the size of \mathcal{D} , i.e., the number of hyperboxes used for the treatment decision. The parameter M can be used to control the complexity of the policy and, thus, also the degree of interpretability. Note that the integrality of ξ_i follows by the integrality constraint of s_j . Hence, we dropped the extra constraint.

4.3. Theoretical Analysis of Π_H^M

In this section, we derive theoretical approximation guarantees for our policy class Π_H^M .

²Note that an optimal union of hyperboxes with respect to (EBCP) can always be built from vertices of the hyperboxes spanned by the subsets of the covariates $\{X_i\}_{i=1}^n$. We can thus set $\mathcal{K} = \{S = [l_1, u_1] \times [l_2, u_2] \times \dots \times [l_d, u_d] : l_t = \min_{i \in \mathcal{I}} X_{i,t} \text{ and } u_t = \max_{i \in \mathcal{I}} X_{i,t} \text{ with } \mathcal{I} \subseteq \{1, \dots, n\}\}$

4.3.1. UNIVERSAL APPROXIMATION THEOREM

The approximation quality is strongly related to the regret, i. e., $\min_{\pi \in \{-1,1\}^{\mathcal{X}}} V(\pi) - V(\pi^{\text{opt}})$. In particular, we prove in Theorem 4.1 that, if an optimal policy $\pi^* \in \arg \min_{\pi \in \{-1,1\}^{\mathcal{X}}} V(\pi)$ is a measurable function, it can be approximated arbitrarily well within our policy class Π_H^M .

Theorem 4.1. *Let $1 \leq p < \infty$ and $\pi^* : \mathcal{X} \rightarrow \{-1, 1\}$ be any Lebesgue measurable function³. Then, for every $\delta \in (0, 1)$ and $\epsilon > 0$, there exists a sample size $n_{\delta, \epsilon} \in \mathbb{N}$ and $M \in \mathbb{N}$ sufficiently large, as well as, a policy $\pi_{\mathcal{D}^*} \in \Pi_H^M$ as defined in (3), such that*

$$\|\pi^* - \pi_{\mathcal{D}^*}\|_p < \epsilon \quad (15)$$

holds with probability at least $1 - \delta$.

See Appendix D for a proof. Theorem 4.1 proves that our policy class is sufficiently rich to approximate arbitrarily complex dependencies between patient covariates and treatment decisions.

4.3.2. ASYMPTOTIC ESTIMATION PROPERTIES

Constraint (13) in the definition of (MILP) restricts the policy class to policies with $|\mathcal{D}| \leq M$. From a computational perspective two questions remain: (i) What is the asymptotic behavior of the estimated policy regarding the finite sample size, i. e., does the error $|\mathcal{J}(\pi^{\text{opt}}) - \mathcal{J}(\hat{\pi}_n^{\text{opt}})|$ for $\hat{\pi}_n^{\text{opt}} \in \arg \min_{\pi \in \Pi_H^M} \mathcal{J}_n(\pi)$ converge to zero for n large enough? (ii) What is the role of the parameter M in this analysis, as the complexity of Π_H^M seems to grow with M and the sample size n ? Both questions are addressed in the following.

Theorem 4.2. *The empirical Rademacher complexity⁴ $R_n(\Pi_H^M)$ of Π_H^M can be bounded independently of the sample size n , i. e., it holds*

$$R_n(\Pi_H^M) \leq C\sqrt{2M}, \quad (16)$$

for a constant $C > 0$.

See Appendix E for a proof. Theorem 4.2 directly implies the following Corollary.

Corollary 4.3. *For a fixed parameter M it holds*

$$|\mathcal{J}(\pi^{\text{opt}}) - \mathcal{J}(\hat{\pi}_n^{\text{opt}})| = \mathcal{O}_p(1/\sqrt{n}), \quad (17)$$

where $\hat{\pi}_n^{\text{opt}} \in \arg \min_{\pi \in \Pi_H^M} \mathcal{J}_n(\pi)$.

See Appendix E for a proof. Corollary 4.3 shows that optimizing (EBCP) yields near-optimal solutions for the policy

³For simplicity, we assume \mathcal{X} to have finite Lebesgue measure.

⁴The empirical Rademacher complexity is defined as $R_n(\Pi) = \frac{1}{n} \mathbb{E}_{\sigma} [\sup_{\pi \in \Pi} \sum_{i=1}^n \sigma_i \pi(X_i)]$, where the expectation is taken over i.i.d. samples of the Rademacher distribution, i. e., $\{\sigma_1, \dots, \sigma_n\}$ with $\mathbb{P}(\sigma_i = 1) = \mathbb{P}(\sigma_i = -1) = \frac{1}{2}$.

value optimization problem (2) with $\Pi = \Pi_H^M$. Furthermore, Theorem 4.1 shows that, for any given δ and ϵ , there exists a sample size $n_{\delta, \epsilon} \in \mathbb{N}$ and $M^* = |\mathcal{D}^*|$, such that with probability $1 - \delta$ there exists a policy $\pi_{\mathcal{D}} \in \Pi_H^M$ for every $M \geq M^*$ with $\|\pi^* - \pi_{\mathcal{D}}\|_p < \epsilon$.

4.4. Choice of the Number of Hyperboxes

Evidently, the derivation of M^* is non-constructive. As a remedy, we discuss two approaches to calibrate the number of hyperboxes, i. e., rules, in practice: (i) expert-informed selection and (ii) an expert-informed penalty method.

Expert-informed selection. We expect clinical practitioners to bound the overall number of rules in our model. To do so, we explicitly model the number of hyperboxes as a constraint, see (13). We think that this is natural in our setting in which interpretable policies are preferred. In this way, the accuracy-complexity tradeoff can easily be controlled in practice by an expert-informed hyperparameter grid for M .

Expert-informed penalty method. If it is preferred to determine the number of hyperboxes, i. e., rules, in a fully data-driven manner, we propose a penalty method for (MILP) in Appendix F. In this case, practitioners have to set a value $\omega > 0$ they are willing to sacrifice in the policy value in order to have one fewer rule in the final model (Rudin, 2019). In our case study with real-world data, for instance, Y decodes the increase in the CD4 cell count after 20 weeks. That is, setting for example $\omega = 25$, one is willing to sacrifice a decrease of the population wide CD4 cell count by 25 cells/mm³ for every rule that is eliminated in the final model.

4.5. Derivation of IOPL

The above formulation of (MILP) is impractical for real-world datasets as the number of binary variables grows exponentially in n . As a remedy, we now propose an efficient optimization algorithm for solving (MILP), which we call IOPL. IOPL consists of (i) a branch-and-bound framework that theoretically allows to solve (MILP) to optimality (see Section 4.5.1) and (ii) a column generation procedure that is able to solve the involved relaxations of (MILP) (see Section 4.5.2). Our algorithm thus falls in the wider area of branch-and-price algorithms (Barnhart et al., 1998).

Our optimization algorithm proceeds as follows. Using a branch-and-price framework, we successively solve *relaxations* of (MILP), where constraint (14) is replaced by $s_j, \xi_i \in [0, 1]$ for all i and j . We denote this relaxation by (RMILP). If a solution has a non-integer value in an s_j , a cutting plane is used to branch into two subproblems where each subproblem fixes s_j either to zero or to one, i. e., adds constraints of the form $s_j \leq 0$ or $s_j \geq 1$, for a certain j determined in a so-called branching rule. To solve the

relaxed problem (RMILP) to optimality, most of the hyperboxes of \mathcal{K} are left out as there are too many to handle them efficiently. For this, we restrict the set of hyperboxes \mathcal{K} to a subset $\mathcal{W} \subseteq \mathcal{K}$ and solve the *restricted* problem denoted by $\text{RMILP}(\mathcal{W})$. To check if the solution of the restricted problem is also optimal for the unrestricted problem, we solve the so-called pricing problem. In the pricing problem, we try to identify new hyperboxes $S \in \mathcal{K} \setminus \mathcal{W}$ that one can add to the restricted linear program (by adding it to the set \mathcal{W}). This is done by identifying missing hyperboxes that yield a negative reduced cost. If all hyperboxes that have not been added have a non-negative reduced cost (i. e., the optimal solution of the pricing problem is greater or equal to zero), the solution of the restricted problem $\text{RMILP}(\mathcal{W})$ and the solution of the unrestricted problem (RMILP) = $\text{RMILP}(\mathcal{K})$ coincide. Note that each solution of (RMILP) yields a lower bound of (MILP), which is used to prune suboptimal branches.

4.5.1. BRANCH-AND-BOUND FRAMEWORK FOR IOPL

In this section, we formalize IOPL; see Algorithm 1. We use the set $\mathcal{C} \subseteq \{1, \dots, |\mathcal{W}|\} \times \{1, 0\}$ within $\text{RMILP}(\mathcal{K}, \mathcal{C})$ to denote the added cutting planes, i. e., the indices $j \in \{1, \dots, |\mathcal{W}|\}$ and the type of cut (i. e., $s_j \leq 0$ or $s_j \geq 1$).

Algorithm 1 IOPL

Input: Initial working set \mathcal{W}_0
Output: Optimal subset $\mathcal{K}^* \subseteq \mathcal{K}$, optimal solution s^*
 Initialize the list of active subproblems $\mathcal{L} \leftarrow \{\text{RMILP}(\mathcal{W}_0, \emptyset)\}$
 Initialize iteration counter $l \leftarrow 0$
repeat
 Select and remove first subproblem $\text{RMILP}(\mathcal{W}, \mathcal{C})$ from \mathcal{L}
 Perform column generation to get the current relaxed solution and the new working set $s', v', \mathcal{W}' \leftarrow \text{ColumnGeneration}(\text{RMILP}(\mathcal{W}, \mathcal{C}))$
 if $l = 0$ **then**
 Solve restricted integer problem to get current optimal integer solution and objective value $s^*, v^* \leftarrow \text{Solve}(\text{MILP}(\mathcal{W}'))$
 Update optimal subset $\mathcal{W}^* \leftarrow \mathcal{W}'$
 end if
 if $v' \leq v^*$ **then**
 if s' integral **then**
 Update new optimal integer solution $(\mathcal{W}^*, s^*) \leftarrow (\mathcal{W}', s')$
 else
 Set j' according to branching rule
 Branch by updating $\mathcal{L} \leftarrow \mathcal{L} \cup \{\text{RMILP}(\mathcal{W}', \mathcal{C} \cup \{(j', 1)\}), \text{RMILP}(\mathcal{W}', \mathcal{C} \cup \{(j', 0)\})\}$
 Solve restricted problem $(s', v') \leftarrow \text{Solve}(\text{MILP}(\mathcal{W}'))$
 if $v' \leq v^*$ **then**
 Update new optimal integer solution $(\mathcal{W}^*, s^*) \leftarrow (\mathcal{W}', s')$
 end if
 end if
 end if
 Increase counter $l \leftarrow l + 1$
until \mathcal{L} empty
return \mathcal{W}^*, s^*

Algorithm 1 proceeds as follows. We initialize the list of active subproblems. Then, the algorithm loops through this list until there are no active subproblems left. In each step, we select the first active subproblem from the list and solve it via column generation (see next section for details). If the optimal objective function value is larger than the current best value, we can prune the current branch and continue. If, on the other hand, the objective function value is lower than the current best value, the algorithm checks if the cor-

responding solution is integral or not. In case it is already integral, we update the current solution. If the solution is fractional, we branch according to a specified branching rule. The latter returns an index j' and adds the corresponding two subproblems to the set of active subproblems. In addition to these branch-and-bound steps, we also solve the restricted (MILP), denoted by $\text{MILP}(\mathcal{W})$, after each subproblem solution that is fractional and in the first iteration. This results in a faster convergence to reasonable candidate solutions by allowing for faster pruning.

Customized branching rule. A crucial step in every branch-and-bound algorithm is the branching rule. We first experimented with standard approaches such as most infeasible branching (Achterberg et al., 2005), i. e. $j' \leftarrow \arg \min_j |s'_j - 0.5|$. However, we achieve a faster convergence through the following customized branching rule: We set j' to the index which corresponds to the box $S^{j'}$ with the largest volume among all boxes with fractional $s_{j'}$. In this way, we fix large boxes early in the branching process, which leads to a faster convergence as these boxes usually have the largest effect in the objective function.

4.5.2. SUBPROBLEM SOLUTIONS VIA COLUMN GENERATION

We now show how to solve the involved subproblems in Algorithm 1. For this purpose, we propose an efficient column generation procedure for the subproblems. To do so, we first derive the reduced cost for a potential new hyperbox S . For this, let $\mu_i^1 \geq 0$ for $i \in I_1 \cap \mathcal{P}$, $\mu_{i,j}^2 \geq 0$ for $i \in I_{-1} \cap \mathcal{P}$ and $j \in \mathcal{K}_i$, $\mu_{i,j}^3 \geq 0$ for $i \in I_1 \cap \mathcal{N}$ and $j \in \mathcal{K}_i$, and $\mu_i^4 \geq 0$ for $i \in I_{-1} \cap \mathcal{N}$ denote the dual variables associated with constraints (9) to (12) of $\text{RMILP}(\mathcal{W})$. Furthermore, let $\lambda \geq 0$ be the dual variable associated with constraint (13). The reduced cost for a new potential hyperbox S is given by

$$\begin{aligned}
 & - \sum_{i \in I_1 \cap \mathcal{P}} \mu_i^1 \delta_i + \sum_{i \in I_{-1} \cap \mathcal{P}} \left(\sum_{j \in \mathcal{K}_i} \mu_{i,j}^2 \right) \delta_i \\
 & + \sum_{i \in I_1 \cap \mathcal{N}} \left(\sum_{j \in \mathcal{K}_i} \mu_{i,j}^3 \right) \delta_i - \sum_{i \in I_{-1} \cap \mathcal{N}} \mu_i^4 \delta_i + \lambda,
 \end{aligned} \tag{18}$$

where $\delta_i \in \{0, 1\}$ denotes whether $X_i \in S$ holds. The first four terms are the sums of the dual variables associated with the respective constraints in which the potential hyperbox appears. Then, the pricing problem consists of minimizing (18) over all potential hyperboxes.

Based on the idea of column generation, we now successively solve the pricing problem. If the minimal value is greater or equal to zero, there is no hyperbox that can be added with a negative reduced cost and, hence, the column generation procedure terminates. If the minimal value is smaller than zero, we (i) add the corresponding hyperbox to the working set \mathcal{W} , (ii) solve $\text{RMILP}(\mathcal{W})$ for the new working set \mathcal{W} to update the dual variables, and (iii) solve

again the pricing problem. In this manner, the working set \mathcal{W} grows until the solution of RMILP(\mathcal{W}) and (RMILP) coincide. Note also that the reduced cost is independent of additionally added cutting planes and, hence, can be used to generate new columns independently of the set \mathcal{C} .

One observes that (18) is a linear combination of δ_i with positive and negative weights. Determining a hyperbox that maximizes such an objective is called hyperbox search in the literature (Louveaux & Mathieu, 2014). We assume w.l.o.g. that all coefficients (i. e., dual variables or sums of dual variables) are strictly larger than zero (otherwise they can be ignored in the corresponding summation) and define the pricing problem as in (Louveaux & Mathieu, 2014) by

$$\max_{\delta, \gamma, p, q} \sum_{i \in I_1 \cap \mathcal{P}} \mu_i^1 \delta_i - \sum_{i \in I_{-1} \cap \mathcal{P}} \left(\sum_{j \in \mathcal{K}_i} \mu_{i,j}^2 \right) \delta_i \quad (19)$$

$$- \sum_{i \in I_1 \cap \mathcal{N}} \left(\sum_{j \in \mathcal{K}_i} \mu_{i,j}^3 \right) \delta_i + \sum_{i \in I_{-1} \cap \mathcal{N}} \mu_i^4 \delta_i - \lambda \quad (20)$$

$$\text{s.t. } \delta_i \leq \gamma_t^{r_t^i} \text{ for all } i \in (I_1 \cap \mathcal{P}) \cup (I_{-1} \cap \mathcal{N}) \text{ and } t \in \{1, \dots, d\}, \quad (21)$$

$$\delta_i + \sum_{t=1}^d (1 - \gamma_t^{r_t^i}) \geq 1 \text{ for all } i \in (I_{-1} \cap \mathcal{P}) \cup (I_1 \cap \mathcal{N}), \quad (22)$$

$$\gamma_t^k = p_t^k - q_t^k \text{ for all } t \in \{1, \dots, d\} \text{ and } k \in \{1, \dots, n_t\}, \quad (23)$$

$$p_t^k \leq p_t^{k+1} \text{ for all } t \in \{1, \dots, d\} \text{ and } k \in \{1, \dots, n_t - 1\}, \quad (24)$$

$$q_t^k \leq q_t^{k+1} \text{ for all } t \in \{1, \dots, d\} \text{ and } k \in \{1, \dots, n_t - 1\}, \quad (25)$$

$$\delta_i, \gamma_t^k, p_t^k, q_t^k \in \{0, 1\}. \quad (26)$$

Thereby, the following definitions and assumptions have been made. In each dimension $t \in \{1, \dots, d\}$, we sort the covariates of patients in increasing order, i. e., sort the set $\{X_{i,t}\}_{i=1}^n$. If some patients have the same covariate in dimension t , we remove it from the set. This results in a list of n_t distinct and sorted values for each dimension. For a patient i , we then define the index in that list in dimension t by r_t^i . Furthermore, let γ_t^k denote whether or not the k -th value in this list lies within the projected interval of the candidate hyperbox onto dimension t . Now, constraint (21) denotes that an individual with positive coefficient, i. e., $i \in (I_1 \cap \mathcal{P}) \cup (I_{-1} \cap \mathcal{N})$, can be selected if the respective covariate lies within the projected interval of the candidate hyperbox in each dimension. Constraint (22) says that a patient with negative coefficient, i. e., $i \in (I_{-1} \cap \mathcal{P}) \cup (I_1 \cap \mathcal{N})$, can be excluded if, in at least one dimension, the covariate lies outside the projected interval of the candidate hyperbox. Constraints (23) to (25) encode the consecutive property of γ_t^k with auxiliary variables p_t^k and q_t^k ; see (Louveaux & Mathieu, 2014) for details. Finally, the resulting box can be built from the values of δ , i. e., for each dimension t we set $l_t = \min_{\{i: \delta_i=1\}} X_{i,t}$ and $u_t = \max_{\{i: \delta_i=1\}} X_{i,t}$.

Computational complexity. The computational complexity of IOPL is mainly driven by the cost for solving the pricing problem. While this is known to be NP-hard (Lou-

veaux & Mathieu, 2014), we find that we can efficiently solve real-world problem instance with $n \approx 1,000$, which is a typical problem size in clinical practice (see Section 5.2). One reason might be that a large part of the dual variables μ are zero during optimization.

5. Experiments

In this section, we investigate IOPL in a variety of settings. Using simulated data, we illustrate how IOPL approximates the optimal decision regions and compare the regret against state-of-the-art methods from *interpretable* off-policy learning. The latter shows that IOPL yields regrets that are substantially lower than state-of-the-art methods. Furthermore, IOPL performs on par with state-of-the-art “black box” methods in off-policy learning. Using real-world clinical data, we assess the interpretability of our policies. For this, we conduct a user study with clinical experts to rate our policies in terms of interpretability, which demonstrates that IOPL indeed yields policies that are perceived as being interpretable in practice.

Baselines. We consider three baselines from state-of-the-art interpretable off-policy learning: (i) *Linear eligibility score* (ES): For this baseline, we set Π to the class of linear functions decoding a treatment ($T = 1$) if the value is greater or equal than zero and no treatment ($T = -1$) otherwise (Kitagawa & Tetenov, 2018). (ii) *Decision lists* (DL): For this baseline, we consider decision lists as in (Zhang et al., 2018). (iii) *Tree-based policy learner* (DT): For this baseline, we consider a tree-based approach with doubly robust correction techniques (Zhang et al., 2012). All baselines are considered as *interpretable* (Rudin, 2019).

For comparison, we also report the performance of a “black box” model, however, for which interpretability is precluded. We consider the class of *deep neural networks* (DNN), which allows for complex decision boundaries but is considered non-intelligible (Rudin, 2019). This allows to give a lower bound on the performance, when dropping the interpretability requirement from clinical practice. Details on all baselines and hyperparameters are given in Appendix G.

Performance. We evaluate the performance in our simulation study by reporting the regret, defined as $\min_{\pi \in \{-1, 1\}^x} V(\pi) - V(\hat{\pi}^{\text{opt}})$. To do so, we sample i.i.d. observations and use a subset for training. Afterward, the regret is computed on all sampled observations. We use the exact nuisance functions μ_t and e_t in the simulation study for all baselines and IOPL. For the real-world clinical data, we use a kernel regression for μ_t and logistic regression for e_t .

Implementation of IOPL. We provide a publicly available implementation of IOPL in Python. For solving the LP (linear program) relaxations and the pricing problem, we

use Gurobi 9.0. We stop IOPL if it exceeds $l = 50$ branch-and-bound iterations as given in Algorithm 1. We set a maximum time limit of 180 seconds for solving the pricing problem in our experiments. We emphasize that this time limit was never exceeded in our experiments, including the experiments with the real-world clinical data (see Section 4.5.2 for a discussion of the reasons).⁵

Hardware. We run all of our experiments on a server with two 16-core Intel Xeon Gold 6242 processors each with 2.8GHz and 192GB of RAM.

5.1. Simulation Study

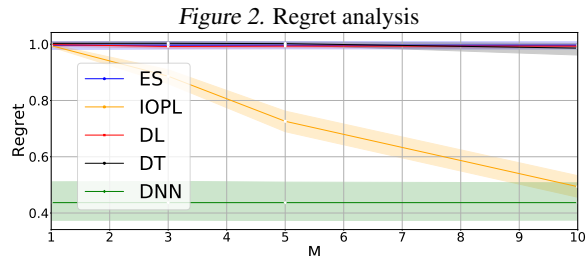
We perform a simulation study to further investigate IOPL. In particular, our objective is to (i) illustrate how our policies approximate the optimal decision region and (ii) compare the performance to state-of-the-art interpretable off-policy learning methods.

Illustration. For illustrative purposes, we generate 2-dimensional patient covariates $X_0, X_1 \sim U[-1, 1]$, a treatment assignment which is independent of X with $\mathbb{P}(T = 1) = 1/2$, and an outcome $Y \mid X, T \sim \mathcal{N}(m(X, T), \sigma^2)$, where $m(X, T) = 1 + 2X_0 + X_1 + f(X, T)$ and $\sigma = 0.1$. The experimental setup is based on Zhao et al. (2012). The benefit of a 2-dimensional setting is that it allows to visualize the true decision regions. The function f introduces interactions between treatments and the outcome variable. We investigate three different scenarios of increasing complexity, see Appendix H for detailed definitions. The first scenario (“basic”) corresponds to a parabola as the decision boundary. The second scenario (“complex”) describes a complex decision boundary for which it is optimal to treat individuals with covariates inside a ring. The third scenario (“very complex”) corresponds to the most complex scenario with two disconnected decision regions and highly nonlinear decision boundaries.

In Figure 3 from Appendix H.1, we observe that, for smaller numbers of hyperboxes, i.e. smaller M , IOPL yields a coarse covering of the decision region, while an increasing number of hyperboxes yields a finer approximation of the decision region. We also report the resulting regrets for all three methods (DR, DM, and IPS) in Appendix H.2, in which the doubly robust approach performs best. Thus, we restrict the following experiments to the use of the doubly robust method. This goes in line with earlier research (Bennett & Kallus, 2020).

Regret analysis. In order to compare the performance of IOPL against our baselines, we generate 4-dimensional patient covariates $X_0, X_1, X_2, X_3 \sim U[-1, 1]$, a treatment assignment which is independent of X with $\mathbb{P}(T = 1) = 1/2$,

⁵Code available at <https://github.com/DanielTschernutter/IOPL>



Notes. We sample 10 different datasets using the described data-generating processes. For IOPL, DL, and DT, we vary the parameter M in $\{1, 3, 5, 10\}$; see Appendix G for details. The shaded areas are the 95% confidence intervals.

and an outcome $Y \mid X, T \sim \mathcal{N}(m(X, T), \sigma^2)$ where $m(X, T) = \max(X_2 + X_3, 0) + 1/2 T \text{sign}(\prod_{i=0}^3 X_i)$, and $\sigma = 0.1$. The experimental setup is based on Athey & Wager (2021). See Appendix J for details.

The results are reported in Figure 2. First, IOPL outperforms all baselines from interpretable off-policy learning. Second, the decision list and tree-based baselines perform only slightly better than the eligibility score, i.e., the linear baseline. This is in line with the fact that both do not offer theoretical approximation guarantees. For additional experiments with a simpler data generating process, see Appendix I. Third, for larger values of M , IOPL yields a regret that is on par with the “black box” method (DNN).

5.2. Real-World Clinical Data

We further assess the interpretability of our policies in a user study based on real-world clinical data. We draw upon the AIDS Clinical Trial Group (ACTG) study 175 (Hammer et al., 1996). The ACTG 175 study assigned treatments randomly to patients with human immunodeficiency virus (HIV) type 1. We consider two treatment arms: (i) both zidovudine (ZDV) and zalcitabine (ZAL) ($T = 1$) vs. (ii) ZDV only ($T = -1$). This corresponds to $n = 1,056$ patients. We consider $d = 12$ covariates similar to (Lu et al., 2013). Further details on the ACTG study are in Appendix K. We trained ES, as well as DT and IOPL, with M varying in $\{1, 3, 5\}$. For $M = 5$, IOPL returned the policy in Figure 1. As we use real clinical data, we do not have access to the true policy value, which impedes a regret analysis in this setting. For the sake of completeness, the empirical policy values can be found in Appendix L.

User study. In a user study, we asked 10 practitioners to rate the interpretability of our policy and the baseline policies on a scale between 0 (black box) and 10 (fully transparent) and whether they would use our policies in practice (see Appendix M). All but one practitioner perceived our policies as interpretable (i.e., a rating larger than 5, which denotes neutral). Furthermore, 9 out of 10 would consider using

our treatment decisions in practice. This demonstrates that our policies fulfill interpretability demands from clinical practice. A detailed comparison to our baselines can be found in Appendix M.

6. Conclusion

We propose IOPL, a novel algorithm for interpretable off-policy learning. It is based on a policy class Π_H^M that assumes a representation of the decision region as a union of hyperboxes. As a result, our treatment decisions are in disjunctive normal form. To optimize over Π_H^M , we develop a tailored branch-and-price framework that allows for efficient training. In addition, we prove that our policy class is sufficiently rich and thus flexible enough to approximate any measurable policy arbitrarily well.

7. Acknowledgements

We acknowledge funding from the Swiss National Science Foundation (Grant 186932).

References

- Achterberg, T., Koch, T., and Martin, A. Branching rules revisited. *Operations Research Letters*, 33(1):42–54, 2005.
- Ahmad, M. A., Eckert, C., and Teredesai, A. Interpretable machine learning in healthcare. *International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pp. 559–560, 2018.
- Athey, S. and Wager, S. Policy learning with observational data. *Econometrica*, 89(1):133–161, 2021.
- Athey, S., Wager, S., et al. Efficient policy learning. *arXiv preprint arXiv:1702.02896*, 78, 2017.
- Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P., and Vance, P. H. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.
- Battocchi, K., Dillon, E., Hei, M., Lewis, G., Oka, P., Oprescu, M., and Syrgkanis, V. EconML: A Python Package for ML-Based Heterogeneous Treatment Effects Estimation. <https://github.com/microsoft/EconML>, 2019. Version 0.12.0.
- Bennett, A. and Kallus, N. Efficient policy learning from surrogate-loss classification reductions. *International Conference on Machine Learning (ICML)*, pp. 788–798, 2020.
- Bertsimas, D., Dunn, J., and Mundru, N. Optimal prescriptive trees. *INFORMS Journal on Optimization*, 1(2): 164–183, 2019.
- Beygelzimer, A. and Langford, J. The offset tree for learning with partial labels. *International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 129–138, 2009.
- Bi, J., Zhang, T., and Bennett, K. P. Column-generation boosting methods for mixture of kernels. *International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 521–526, 2004.
- Blatt, D., Murphy, S. A., and Zhu, J. A-learning for approximate planning. *Ann Arbor*, 1001:48109–2122, 2004.
- Blümlein, T., Persson, J., and Feuerriegel, S. Learning optimal dynamic treatment regimes using causal tree methods in medicine. *arXiv preprint arXiv:2204.07124*, 2022.
- Bottou, L., Peters, J., Quiñero-Candela, J., Charles, D. X., Chikering, D. M., Portugaly, E., Ray, D., Simard, P., and Snelson, E. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research*, 14(65):3207–3260, 2013.
- Chan, I. S. and Ginsburg, G. S. Personalized medicine: progress and promise. *Annual Review of Genomics and Human Genetics*, 12:217–244, 2011.
- Collins, F. S. and Varmus, H. A new initiative on precision medicine. *The New England Journal of Medicine*, 372(9):793–795, 2015.
- Dash, S., Günlük, O., and Wei, D. Boolean decision rules via column generation. *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 4655–4665, 2018.
- Demiriz, A., Bennett, K. P., and Shawe-Taylor, J. Linear programming boosting via column generation. *Machine Learning*, 46(1/3):225–254, 2002.
- Doshi-Velez, F. and Kim, B. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608v2*, 2017.
- Dudík, M., Erhan, D., Langford, J., and Li, L. Doubly robust policy evaluation and optimization. *Statistical Science*, 29(4), 2014.
- Hamburg, M. A. and Collins, F. S. The path to personalized medicine. *The New England Journal of Medicine*, 363(4):301–304, 2010.
- Hammer, S. M., Katzenstein, D. A., Hughes, M. D., Gundacker, H., Schooley, R. T., Haubrich, R. H., Henry, W. K., Lederman, M. M., Phair, J. P., Niu, M., Hirsch, M. S., and Merigan, T. C. A trial comparing nucleoside monotherapy with combination therapy in hiv-infected adults with cd4 cell counts from 200 to 500 per cubic millimeter. *aids clinical trials group study 175 study team*.

- The New England Journal of Medicine*, 335(15):1081–1090, 1996.
- Horvitz, D. G. and Thompson, D. J. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260):663–685, 1952.
- Imbens, G. and Rubin, D. B. *Causal inference for statistics, social, and biomedical sciences: An introduction*. Cambridge University Press, Cambridge, 2015.
- Interpretable AI, L. Interpretable AI Documentation, 2022. URL <https://www.interpretable.ai>.
- Jiang, B., Song, R., Li, J., and Zeng, D. Entropy learning for dynamic treatment regimes. *Statistica Sinica*, 29(4):1633–1655, 2019.
- Kallus, N. Balanced policy evaluation and learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.
- Kallus, N. and Zhou, A. Confounding-robust policy improvement. *arXiv preprint arXiv:1805.08593*, 2018.
- Kattan, M. W., Hess, K. R., Amin, M. B., Lu, Y., Moons, K. G. M., Gershengwald, J. E., Gimotty, P. A., Guinney, J. H., Halabi, S., Lazar, A. J., Mahar, A. L., Patel, T., Sargent, D. J., Weiser, M. R., and Compton, C. American joint committee on cancer acceptance criteria for inclusion of risk models for individualized prognosis in the practice of precision medicine. *CA: A Cancer Journal for Clinicians*, 66(5):370–374, 2016.
- Kitagawa, T. and Tetenov, A. Who should be treated? empirical welfare maximization methods for treatment choice. *Econometrica*, 86(2):591–616, 2018.
- Kosorok, M. R. and Laber, E. B. Precision medicine. *Annual Review of Statistics and its Application*, 6:263–286, 2019.
- Laber, E. B. and Zhao, Y.-Q. Tree-based methods for individualized treatment regimes. *Biometrika*, 102(3):501–514, 2015.
- Laber, E. B., Linn, K. A., and Stefanski, L. A. Interactive model building for q-learning. *Biometrika*, 101(4):831–847, 2014.
- Lakkaraju, H. and Rudin, C. Learning cost-effective and interpretable treatment regimes. *Artificial Intelligence and Statistics*, pp. 166–175, 2017.
- Li, L., Chu, W., Langford, J., and Wang, X. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. *International Conference on Web Search and Data Mining (WSDM)*, pp. 297–306, 2011.
- Li, X., Lin, G., Shen, C., Hengel, A., and Dick, A. Learning hash functions using column generation. *International Conference on Machine Learning (ICML)*, pp. 142–150, 2013.
- Louveaux, Q. and Mathieu, S. A combinatorial branch-and-bound algorithm for box search. *Discrete Optimization*, 13:36–48, 2014.
- Lu, W., Zhang, H. H., and Zeng, D. Variable selection for optimal treatment decision. *Statistical Methods in Medical Research*, 22(5):493–504, 2013.
- Murphy, S. A. Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(2):331–355, 2003.
- Pan, Y. and Zhao, Y.-Q. Improved doubly robust estimation in learning optimal individualized treatment rules. *Journal of the American Statistical Association*, 116(533):283–294, 2021.
- Qian, M. and Murphy, S. A. Performance guarantees for individualized treatment rules. *Annals of Statistics*, 39(2):1180–1210, 2011.
- Robins, J. M. Optimal structural nested models for optimal sequential decisions. *Proceedings of the Second Seattle Symposium in Biostatistics*, pp. 189–326, 2004.
- Rubin, D. B. Causal inference using potential outcomes. *Journal of the American Statistical Association*, 100(469):322–331, 2005.
- Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.
- Schulte, P. J., Tsiatis, A. A., Laber, E. B., and Davidian, M. Q- and a-learning methods for estimating optimal dynamic treatment regimes. *Statistical Science*, 29(4), 2014.
- Stein, E. M. and Shakarchi, R. *Real Analysis: Measure Theory, Integration, and Hilbert Spaces*. Princeton University Press, 2009.
- Thomas, P. and Brunskill, E. Data-efficient off-policy policy evaluation for reinforcement learning. *International Conference on Machine Learning (ICML)*, pp. 2139–2148, 2016.
- Wang, T., Rudin, C., Doshi-Velez, F., Liu, Y., Klampfl, E., and MacNeille, P. A bayesian framework for learning rule sets for interpretable classification. *Journal of Machine Learning Research*, 18(1):2357–2393, 2017.

- Zhang, B., Tsiatis, A. A., Davidian, M., Zhang, M., and Laber, E. Estimating optimal treatment regimes from a classification perspective. *Stat*, 1(1):103–114, 2012.
- Zhang, Y., Laber, E. B., Tsiatis, A., and Davidian, M. Using decision lists to construct interpretable and parsimonious treatment regimes. *Biometrics*, 71(4):895–904, 2015.
- Zhang, Y., Laber, E. B., Davidian, M., and Tsiatis, A. A. Interpretable dynamic treatment regimes. *Journal of the American Statistical Association*, 113(524):1541–1549, 2018.
- Zhao, Y., Zeng, D., Rush, A. J., and Kosorok, M. R. Estimating individualized treatment rules using outcome weighted learning. *Journal of the American Statistical Association*, 107(499):1106–1118, 2012.
- Zhao, Y.-Q., Laber, E. B., Ning, Y., Saha, S., and Sands, B. E. Efficient augmentation and relaxation learning for individualized treatment rules using observational data. *Journal of Machine Learning Research*, 20(48): 1–23, 2019.
- Zhou, X., Mayer-Hamblett, N., Khan, U., and Kosorok, M. R. Residual weighted learning for estimating individualized treatment rules. *Journal of the American Statistical Association*, 112(517):169–187, 2017.
- Zhou, Z., Athey, S., and Wager, S. Offline multi-action policy learning: Generalization and optimization. *arXiv preprint arXiv:1810.04778*, 2018.

A. Interpretability of IOPL

In this section, we review IOPL in terms of interpretability with respect to the *AAAI 2021 tutorial* on “Explaining Machine Learning Predictions: State-of-the-art, Challenges, and Opportunities”.⁶

- **Interpretability.** IOPL can be viewed as an inherently interpretable predictive model. This is in contrast to post hoc explainability approaches, where the goal is to explain decisions from black box models in retrospect to benefit from their predictive performance. To the contrary, IOPL offers an interpretable machine learning approach with universal approximation guarantees itself. That is, black box models usually offer better performance, however finding an interpretable representation afterwards still goes along with an increase in the regret due to a lack of flexibility (for instance, when explaining neural networks with fixed depth decision trees). As a remedy, our approach combines both worlds with an easy to use and understandable parameter M , which controls the degree of interpretability.
- **Global vs. local explainability.** As already mentioned in the main paper, IOPL offers policies in disjunctive normal form that allow for global explainability. That is, a comprehensible representation of the treatment groups as in Figure 1 allows clinical experts to critically assess the learned policy, i. e., the complete behavior of the model. If the parameter M is set to very large numbers, a compact representation as in Figure 1 might not be possible anymore. Nevertheless, IOPL still offers local explainability, as individual predictions can be explained by displaying the single hyperbox that led to the corresponding decision. In case that one patient fulfills multiple conditions, i. e., the covariates lie in multiple hyperboxes, either all or the one with the largest volume can be considered. In that way, clinical practitioners can still review if the treatment decision is made on a reasonable basis.
- **Utility.** IOPL yields policies that can help to critically assess the behavior of the model in the following ways: (i) It allows for debugging as medical professionals are offered a comprehensible representation of the model behavior. (ii) Biases can be detected and ruled out, as our policies allow to quickly recognize if, for instance, only men are treated with a certain medication. (iii) The treatment decisions of IOPL are completely transparent and, hence, allows medical practitioners to decide when to trust the model predictions and when they are suitable for deployment.

⁶AAAI 2021 tutorial on “Explaining Machine Learning Predictions: State-of-the-art, Challenges, and Opportunities”, <https://explainml-tutorial.github.io/aaai21>, last accessed 01/25/22.

B. Details on Binary Classification Reformulation

We now show that problem (2) and problem (4) are equivalent. For this, we proceed as follows.

First, we recognize that (2) is equivalent to

$$\pi^{\text{opt}} \in \arg \min_{\pi \in \Pi} V(\pi) - \frac{1}{2} \mathbb{E}[Y(+1) - Y(-1)], \quad (27)$$

since we only subtract a constant. Similar to (Bennett & Kallus, 2020), we see that the above problem is equivalent to

$$\pi^{\text{opt}} \in \arg \min_{\pi \in \Pi} \mathbb{E}[\psi \pi(X)], \quad (28)$$

where ψ corresponds to one of the three standard methods for policy learning: direct method ψ^{DM} , inverse propensity score weighted method ψ^{IPS} , and doubly robust method ψ^{DR} defined as

$$\psi^{\text{DM}} = \mu_1(X) - \mu_{-1}(X), \quad \psi^{\text{IPS}} = \frac{TY}{e_T(X)}, \quad \psi^{\text{DR}} = \psi^{\text{DM}} + \psi^{\text{IPS}} - \frac{T\mu_T(X)}{e_T(X)}. \quad (29)$$

Now, we have that

$$\mathbb{E}[\psi \pi(X)] = \mathbb{E}[\psi T^2 \pi(X)] = \mathbb{E}[-\psi T(1 - T\pi(X)) + \psi T], \quad (30)$$

and

$$1 - T\pi(X) = 2I(T \neq \pi(X)). \quad (31)$$

Hence,

$$\mathbb{E}[\psi \pi(X)] = \mathbb{E}[-2T\psi I(T \neq \pi(X))] + \mathbb{E}[\psi T], \quad (32)$$

and thus

$$\pi^{\text{opt}} \in \arg \min_{\pi \in \Pi} \mathbb{E}[\psi \pi(X)] \Leftrightarrow \pi^{\text{opt}} \in \arg \min_{\pi \in \Pi} \mathbb{E}[-T\psi I(T \neq \pi(X))], \quad (33)$$

which proves the claim as $-T\psi$ with ψ given as above corresponds exactly to the definitions of ψ in the main paper.

C. Equivalence of (MILP) and (EBCP)

The empirical binary classification problem (EBCP) is given by

$$\min_{\pi \in \Pi} \frac{1}{n} \sum_{i=1}^n \psi_i I(T_i \neq \pi(X_i)). \quad (34)$$

By setting $\Pi = \Pi_H^M$ and $\xi_i = I(T_i \neq \pi_{\mathcal{D}}(X_i))$ in problem (34), we have to model: (i) the form of the policy $\pi_{\mathcal{D}}$, i. e., which of the N hyperboxes belongs to the policy and (ii) the indicator variables ξ_i given a specific policy $\pi_{\mathcal{D}}$, i. e., a specific choice of hyperboxes.

First, the objective becomes

$$\min_{s, \xi} \frac{1}{n} \sum_{i=1}^n \psi_i \xi_i. \quad (35)$$

For (i), we introduce binary decision variables $s_j \in \{0, 1\}$ for $j \in \{1, \dots, N\}$ indicating whether or not box j is used in the optimal policy. To ensure that the maximal number of hyperboxes M is not exceeded, we use the constraint

$$\sum_{j=1}^N s_j \leq M, \quad (36)$$

which encodes that the number of used hyperboxes is bounded by M .

For (ii), we proceed via a case distinction on the samples i .

Case 1: $i \in I_1 \cap \mathcal{P}$

In this case, patient i was treated, i. e., $T_i = 1$, and the corresponding weight $\Psi_i > 0$. From $T_i = 1$, we get that

$$\xi_i = I(1 \neq \pi_{\mathcal{D}}(X_i)) = \begin{cases} 1, & \text{if } \pi_{\mathcal{D}}(X_i) = -1, \\ 0, & \text{if } \pi_{\mathcal{D}}(X_i) = 1. \end{cases} \quad (37)$$

Furthermore, from $\Psi_i > 0$, we get that, in an optimal solution, ξ_i will strive to zero whenever possible (see objective in (35)). Thus, we need a constraint that forces ξ_i to one if patient i is not treated according to the optimal policy to fulfill (37), and is switched off otherwise to allow ξ_i to strive to zero. This is exactly encoded in the constraint

$$\xi_i + \sum_{j \in \mathcal{K}_i} s_j \geq 1. \quad (38)$$

Note that \mathcal{K}_i is the set of all indices of hyperboxes containing the covariates of patient i . That is, if on the one hand no hyperbox that contains the covariates of patient i is selected we get

$$\sum_{j \in \mathcal{K}_i} s_j = 0, \quad (39)$$

and, thus, constraint (38) becomes $\xi_i \geq 1$, yielding $\xi_i = 1$. If, on the other hand, a hyperbox is selected that contains the covariates of patient i , we get

$$\sum_{j \in \mathcal{K}_i} s_j \geq 1, \quad (40)$$

and, thus, that constraint (38) is fulfilled for any ξ_i , yielding $\xi_i = 0$.

Case 2: $i \in I_1 \cap \mathcal{N}$

In this case, ξ_i is again given by (37). However, now we have that $\Psi_i < 0$, and, thus, we get that, in an optimal solution, ξ_i will strive to one whenever possible (see objective in (35)). Thus, we need a constraint that forces ξ_i to zero if patient i is treated according to the optimal policy to fulfill (37), and is switched off otherwise to allow ξ_i to strive to one. This is encoded in the constraint

$$\xi_i \leq 1 - s_j \text{ for } j \in \mathcal{K}_i. \quad (41)$$

Note that if on the one hand any hyperbox containing the covariates of patient i is selected constraint (41) enforces $\xi_i \leq 0$ and, thus, $\xi_i = 0$. If, on the other hand, no hyperbox that contains the covariates of patient i is selected, constraint (41) becomes $\xi_i \leq 1$ and is thus fulfilled for any ξ_i , yielding $\xi_i = 1$.

The cases $i \in I_{-1} \cap \mathcal{P}$ and $i \in I_{-1} \cap \mathcal{N}$ follow the exact same logic with the difference that we have

$$\xi_i = I(-1 \neq \pi_{\mathcal{D}}(X_i)) = \begin{cases} 0, & \text{if } \pi_{\mathcal{D}}(X_i) = -1, \\ 1, & \text{if } \pi_{\mathcal{D}}(X_i) = 1, \end{cases} \quad (42)$$

in this case.

Again, we note that the integrality of ξ_i follows by the integrality constraint of s_j . That is, due to the objective and our constraints, the values of ξ_i always strive to the extreme points of the interval $[0, 1]$. Hence, the constraint $\xi_i \in [0, 1]$ is sufficient and reduces the number of integer variables significantly compared to $\xi_i \in \{0, 1\}$ for all $i \in \{1, \dots, n\}$.

D. Universal Approximation Theorem

D.1. Proof of Theorem 4.1

First, we observe that

$$\pi^*(x) = \begin{cases} 1 & \text{if } x \in \mathcal{A}, \\ -1 & \text{if } x \in \mathcal{A}^c. \end{cases} \quad (43)$$

As π^* is assumed to be Lebesgue measurable, we have that the decision region \mathcal{A} is Lebesgue measurable with $\lambda(\mathcal{A}) < \infty$, where λ denotes the Lebesgue measure. Note that this implies $\pi^* \in L^p(\mathcal{X})$ for all $1 \leq p < \infty$ as π^* is a simple function.

Now, from Theorem 3.4 in (Stein & Shakarchi, 2009), we know that, for every $\epsilon > 0$, there exists a finite union of boxes $\tilde{\mathcal{A}} = \bigcup_{j=1}^m \tilde{S}^j$ such that

$$\lambda(\mathcal{A} \Delta \tilde{\mathcal{A}}) \leq \frac{\epsilon}{4}, \quad (44)$$

where $\mathcal{A} \Delta \tilde{\mathcal{A}}$ denotes the symmetric difference between two sets, i. e., $\mathcal{A} \Delta \tilde{\mathcal{A}} = \mathcal{A} \setminus \tilde{\mathcal{A}} \cup \tilde{\mathcal{A}} \setminus \mathcal{A}$.

For the next part of the proof, we need a technical lemma (the proof is given in Appendix D.2) given as follows.

Lemma D.1 (Technical lemma). *For every $\delta \in (0, 1)$, $\epsilon > 0$, and box \tilde{S} , there exists a sample size $n_{\delta, \epsilon} \in \mathbb{N}$ and an i.i.d. sample $\{(X_i, T_i, Y_i)\}_{i=1}^{n_{\delta, \epsilon}}$ with $(X_i, T_i, Y_i) \sim (X, T, Y)$, such that there exists a box $\tilde{\tilde{S}}$ generated by a subset $\{X_i\}_{i \in \mathcal{I}} \subseteq \{X_i\}_{i=1}^{n_{\delta, \epsilon}}$, i. e.,*

$$\tilde{\tilde{S}} = [\tilde{l}_1, \tilde{u}_1] \times [\tilde{l}_2, \tilde{u}_2] \times \cdots \times [\tilde{l}_d, \tilde{u}_d] \text{ with } \tilde{l}_t = \min_{i \in \mathcal{I}} X_{i,t} \text{ and } \tilde{u}_t = \max_{i \in \mathcal{I}} X_{i,t}, \quad (45)$$

that fulfills

$$\lambda(\tilde{S} \Delta \tilde{\tilde{S}}) \leq \frac{\epsilon}{4m}, \quad (46)$$

with probability at least $1 - \delta$.

By iteratively applying Lemma D.1, there exists an $n_{\delta, \epsilon} \in \mathbb{N}$ large enough and boxes $\tilde{\tilde{S}}^j$, such that

$$\lambda(\tilde{S}^j \Delta \tilde{\tilde{S}}^j) \leq \frac{\epsilon}{4m}, \quad (47)$$

for all $j \in \{1, \dots, m\}$ with probability at least $1 - \delta$. Now let $\tilde{\tilde{\mathcal{A}}} = \bigcup_{j=1}^m \tilde{\tilde{S}}^j$. We yield

$$\lambda(\mathcal{A} \Delta \tilde{\tilde{\mathcal{A}}}) = \lambda\left(\mathcal{A} \Delta \bigcup_{j=1}^m \tilde{\tilde{S}}^j\right) \quad (48)$$

$$\leq \lambda\left(\mathcal{A} \Delta \bigcup_{j=1}^m \tilde{S}^j\right) + \lambda\left(\bigcup_{j=1}^m \tilde{S}^j \Delta \bigcup_{j=1}^m \tilde{\tilde{S}}^j\right) \quad (49)$$

$$\leq \frac{\epsilon}{4} + \sum_{j=1}^m \lambda(\tilde{S}^j \Delta \tilde{\tilde{S}}^j) \quad (50)$$

$$\leq \frac{\epsilon}{2}, \quad (51)$$

where Equation (49) follows by a generalized triangle inequality for the symmetric difference and Equation (50) follows by the inclusion property of the symmetric difference of unions and the sub-additivity of the Lebesgue measure.

Now, we can write $\pi^* = \chi_{\mathcal{A}} - \chi_{\mathcal{A}^c}$, where χ denotes the characteristic function and define $\pi_{\mathcal{D}^*} = \chi_{\tilde{\mathcal{A}}} - \chi_{\tilde{\mathcal{A}}^c}$ by setting \mathcal{D}^* appropriately and set $M = |\mathcal{D}^*|$. We then yield

$$\|\pi^* - \pi_{\mathcal{D}^*}\|_p = \|\chi_{\mathcal{A}} - \chi_{\tilde{\mathcal{A}}} + \chi_{\tilde{\mathcal{A}}^c} - \chi_{\mathcal{A}^c}\|_p \quad (52)$$

$$\leq \|\chi_{\mathcal{A}} - \chi_{\tilde{\mathcal{A}}}\|_p + \|\chi_{\tilde{\mathcal{A}}^c} - \chi_{\mathcal{A}^c}\|_p. \quad (53)$$

and

$$\|\chi_{\mathcal{A}} - \chi_{\tilde{\mathcal{A}}}\|_p^p = \int_{\mathcal{X}} |\chi_{\mathcal{A}} - \chi_{\tilde{\mathcal{A}}}|^p dx \quad (54)$$

$$= \int_{\mathcal{A} \setminus \tilde{\mathcal{A}}} 1 dx + \int_{\tilde{\mathcal{A}} \setminus \mathcal{A}} 1 dx = \lambda(\mathcal{A} \Delta \tilde{\mathcal{A}}). \quad (55)$$

Analogously, we get

$$\|\chi_{\mathcal{A}^c} - \chi_{\tilde{\mathcal{A}}^c}\|_p^p = \lambda(\mathcal{A}^c \Delta \tilde{\mathcal{A}}^c) = \lambda(\mathcal{A} \Delta \tilde{\mathcal{A}}). \quad (56)$$

Combining the above we thus get

$$\|\pi^* - \pi_{\mathcal{D}^*}\|_p \leq 2\lambda(\mathcal{A} \Delta \tilde{\mathcal{A}})^{\frac{1}{p}} \leq 2\lambda(\mathcal{A} \Delta \tilde{\mathcal{A}}) \leq \epsilon, \quad (57)$$

which proves the claim.

D.2. Proof of Lemma D.1

Let \mathcal{V} be the set of vertices of \tilde{S} . W.l.o.g we assume that $\text{supp}(X) = \mathcal{X}$, otherwise we can redefine \mathcal{X} to contain only X with positive probability. Now, we have that $\mathbb{P}(\|X - v\| < \tilde{\delta}) > 0$ for all $v \in \mathcal{V}$ and $\tilde{\delta} > 0$. That is, when sampling long enough, we will reach a sample with $X_{i_v} \in B_{\tilde{\delta}}(v)$ for all vertices $v \in \mathcal{V}$ and for indices $i_v \in \{1, \dots, n_{\tilde{\delta}, \epsilon}\}$. By choosing $\tilde{\delta}$ and the sample size $n_{\tilde{\delta}, \epsilon}$ appropriately, the claim follows.

E. Asymptotic Estimation Properties

E.1. Proof of Theorem 4.2

To prove Equation (16), we make use of Massart’s Lemma⁷.

Lemma E.1. *Assume $|\Pi|$ is finite. Let $\{(X_i)\}_{i=1}^n$ be a random i.i.d. sample, and let*

$$B = \max_{\pi \in \Pi} \left(\sum_{i=1}^n \pi(X_i)^2 \right)^{\frac{1}{2}}, \quad (58)$$

then

$$R_n(\Pi) \leq \frac{B\sqrt{2\log|\Pi|}}{n}. \quad (59)$$

First, as $\pi \in \Pi_H^M$ are binary valued functions we immediately have that $B = \sqrt{n}$ in our case. Second, we make the following observation

$$|\Pi_H^M| = \sum_{k=0}^M \binom{2^n}{k} \leq \left(\frac{e2^n}{M} \right)^M. \quad (60)$$

By applying Lemma E.1, we yield

$$R_n(\Pi_H^M) \leq \sqrt{\frac{2\log|\Pi_H^M|}{n}} \leq \sqrt{\frac{2M(1+n\log 2 - \log M)}{n}} \quad (61)$$

$$\leq \sqrt{2M + 2M\log 2} = C\sqrt{2M}, \quad (62)$$

where $C = \sqrt{1 + \log 2}$.

E.2. Proof of Corollary 4.3

The proof follows the elaborations in Section 2 in (Bennett & Kallus, 2020). For the sake of completeness, we restate the proof in our setting and notation. (Athey et al., 2017), (Kitagawa & Tetenov, 2018), and (Zhou et al., 2018) gave bounds of the form

$$\sup_{\pi \in \Pi} |\mathcal{J}(\pi) - \mathcal{J}_n(\pi)| = \mathcal{O}_p(1/\sqrt{n}), \quad (63)$$

given that the policy class Π has bounded complexity. Hence, by Theorem 4.2, one yields

$$\mathcal{J}(\pi^{\text{opt}}) - \mathcal{J}(\hat{\pi}_n^{\text{opt}}) \leq \mathcal{J}(\pi^{\text{opt}}) - \mathcal{J}(\hat{\pi}_n^{\text{opt}}) + \mathcal{J}_n(\hat{\pi}_n^{\text{opt}}) - \mathcal{J}_n(\pi^{\text{opt}}) \leq 2 \sup_{\pi \in \Pi} |\mathcal{J}(\pi) - \mathcal{J}_n(\pi)| = \mathcal{O}_p(1/\sqrt{n}). \quad (64)$$

⁷see for instance Theorem 4.3 in http://www.cs.toronto.edu/~rjliao/notes/Notes_on_Rademacher_Complexity.pdf, last accessed 01/25/22.

F. Expert-Informed Penalty Method

To estimate the number of hyperboxes, i. e., the number of rules, in a fully automated manner we propose a penalty method for (MILP) in the following. To do so, we add a penalty term $\omega \sum_{j=1}^N s_j$ to the objective function. Thereby, the penalty parameter $\omega > 0$ can be set by a clinical practitioner and describes the value one is willing to sacrifice in the policy value in order to have one fewer rule in the resulting model (Rudin, 2019). In our case study with real-world data, for instance, Y decodes the increase in the CD4 cell count after 20 weeks. That is, setting for example $\omega = 25$, one is willing to sacrifice a decrease of the population wide CD4 cell count by 25 cells/mm³ for every rule that is eliminated in the final model.⁸ After that, the parameter M can be set to an absolute maximal number, e. g., $M = 100$, and IOPL determines the number of rules automatically.

For our expert-informed penalty method, the framework of IOPL is adapted as follows: (i) we alter (MILP) by adding the discussed penalty term and (ii) we update the reduced cost for the pricing problem accordingly. All other steps in Algorithm 1 remain unchanged.

For (i), (MILP) becomes

$$\min \frac{1}{n} \sum_{i=1}^n \psi_i \xi_i + \omega \sum_{j=1}^N s_j \quad (65)$$

$$\text{s.t. } \xi_i + \sum_{j \in \mathcal{K}_i} s_j \geq 1 \quad \text{for } i \in I_1 \cap \mathcal{P}, \quad (66)$$

$$\xi_i \geq s_j \quad \text{for } i \in I_{-1} \cap \mathcal{P} \text{ and } j \in \mathcal{K}_i, \quad (67)$$

$$\xi_i \leq 1 - s_j \quad \text{for } i \in I_1 \cap \mathcal{N} \text{ and } j \in \mathcal{K}_i, \quad (68)$$

$$\xi_i \leq \sum_{j \in \mathcal{K}_i} s_j \quad \text{for } i \in I_{-1} \cap \mathcal{N}, \quad (69)$$

$$\sum_{j=1}^N s_j \leq M, \quad (70)$$

$$s_j \in \{0, 1\}, \xi_i \in [0, 1]. \quad (71)$$

Note that the penalized objective is still linear. Hence, the computational complexities for the relaxed problem (RMILP) and the restricted (MILP) remain the same.

For (ii), we have to update the reduced cost accordingly. Based on the form of the penalty term it changes merely by an additional additive constant ω , i. e., the reduced cost becomes

$$\begin{aligned} & \omega - \sum_{i \in I_1 \cap \mathcal{P}} \mu_i^1 \delta_i + \sum_{i \in I_{-1} \cap \mathcal{P}} \left(\sum_{j \in \mathcal{K}_i} \mu_{i,j}^2 \right) \delta_i \\ & + \sum_{i \in I_1 \cap \mathcal{N}} \left(\sum_{j \in \mathcal{K}_i} \mu_{i,j}^3 \right) \delta_i - \sum_{i \in I_{-1} \cap \mathcal{N}} \mu_i^4 \delta_i + \lambda. \end{aligned} \quad (72)$$

In summary, the algorithm changes merely in an additional linear term in the objective functions of the relaxed problem RMILP(\mathcal{W}) and the restricted problem MILP(\mathcal{W}). The pricing problem remains unchanged, as we only have to terminate the column generation procedure if the minimal value of the pricing problem is greater or equal to $-\omega$ (instead of zero).

⁸Any scaling of Y must be taken into account, of course.

G. Baselines

G.1. Surrogate-Loss Classification

For the ES and DNN baselines, we follow the literature on off-policy learning and reformulate (EBCP) equivalently via a convex surrogate loss that replaces the discontinuous and non-convex 0-1 loss with the well-established hinge loss (Beygelzimer & Langford, 2009; Jiang et al., 2019; Zhao et al., 2012).

G.1.1. TRAINING

For training, we first note that a straightforward replacement of the 0-1 loss with the hinge-loss in (EBCP) is not possible. This is because the weights ψ_i can be both positive and negative. Hence, the function

$$\frac{1}{n} \sum_{i=1}^n \psi_i \max(1 - T_i \pi(X_i), 0) \quad (73)$$

might not be convex. As a remedy, we rewrite (EBCP) as follows. Note that

$$\psi_i 2I(T_i \neq \pi(X_i)) = \psi_i (1 - T_i \pi(X_i)) = \psi_i - |\psi_i| \text{sign}(\psi_i) T_i \pi(X_i) \quad (74)$$

$$= \psi_i - |\psi_i| + |\psi_i| - |\psi_i| \text{sign}(\psi_i) T_i \pi(X_i) \quad (75)$$

$$= \psi_i - |\psi_i| + |\psi_i| (1 - \text{sign}(\psi_i) T_i \pi(X_i)) \quad (76)$$

$$= \psi_i - |\psi_i| + 2|\psi_i| I(\text{sign}(\psi_i) T_i \neq \pi(X_i)) \quad (77)$$

Thus, (EBCP) can be reformulated, using the hinge loss as a surrogate, as

$$\min_{\pi \in \Pi} \frac{1}{n} \sum_{i=1}^n |\psi_i| \max(1 - \text{sign}(\psi_i) T_i \pi(X_i), 0). \quad (78)$$

To control for over-fitting, we add a regularization term $\rho \text{Reg}(\pi)$ resulting in

$$\min_{\pi \in \Pi} \frac{1}{n} \sum_{i=1}^n |\psi_i| \max(1 - \text{sign}(\psi_i) T_i \pi(X_i), 0) + \rho \text{Reg}(\pi). \quad (\text{H-EBCP})$$

Linear eligibility score. For the linear model, we proceed as follows. We set $\pi(X_i) = \langle w, X_i \rangle - b$ and the regularization term is set to $\text{Reg}(\pi) = \|w\|^2$. Thus, we can rewrite (H-EBCP) as

$$\min_{w \in \mathbb{R}^d, b \in \mathbb{R}, \xi \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n |\psi_i| \xi_i + \rho \|w\|^2 \quad (79)$$

$$\text{s.t. } \xi_i \geq 1 - \text{sign}(\psi_i) T_i (\langle w, X_i \rangle - b) \text{ for all } i \in \{1, \dots, n\}, \quad (80)$$

$$\xi_i \geq 0 \text{ for all } i \in \{1, \dots, n\}. \quad (81)$$

The resulting problem is a weighted linear support vector classification problem that is solved via the LinearSVC class of the sklearn python package.

Deep neural network. For our neural network baseline, we set Reg to the ℓ_2 -regularization. We use PyTorch with a custom loss function and the Adam optimizer to train the neural network parameters. In particular, we use early stopping with a patience of 10 and 20 warmup epochs. The structure of the neural network is summarized in Table 1.

Table 1. Deep Neural Network Structure

Input neurons per layer				Activation
Layer 1	Layer 2	Layer 3	Layer 4	(output-layer)
d	128	128	64	identity

G.2. Decision Lists

For our decision list baseline, we follow (Zhang et al., 2018). That is, we define the set of potential clauses in the decision list as

$$\mathcal{R} = \{\mathcal{X}, \{x \in \mathcal{X} : x_{j_1} \leq \tau_1\}, \{x \in \mathcal{X} : x_{j_1} > \tau_1\}, \quad (82)$$

$$\{x \in \mathcal{X} : x_{j_1} \leq \tau_1 \text{ and } x_{j_2} \leq \tau_2\}, \quad (83)$$

$$\{x \in \mathcal{X} : x_{j_1} \leq \tau_1 \text{ and } x_{j_2} > \tau_2\}, \quad (84)$$

$$\{x \in \mathcal{X} : x_{j_1} > \tau_1 \text{ and } x_{j_2} \leq \tau_2\}, \quad (85)$$

$$\{x \in \mathcal{X} : x_{j_1} > \tau_1 \text{ and } x_{j_2} > \tau_2\} : \quad (86)$$

$$1 \leq j_1 < j_2 \leq d, \tau_1, \tau_2 \in \mathbb{R}\}, \quad (87)$$

where j_1 and j_2 are indices and τ_1 and τ_2 are thresholds. For optimization, we analogously define the optimal estimated decision rule $\hat{\pi}$ using the outcome function as $x \mapsto \arg \min_{t \in \{-1, 1\}} \mu_t(x)$. The estimation then follows Algorithm 2 as proposed in (Zhang et al., 2018).

Algorithm 2 Estimation of Clauses for Decision List Baseline

Input: Maximal number of clauses M , hyperparameter ξ , hyperparameter η

Output: Optimal decision list

for $l \in \{1, \dots, M\}$ **do**

if $l < M$ **then**

 Compute new clause via

$$\begin{aligned} (R_l, t_l) = & \arg \min_{R \in \mathcal{R}, t \in \{-1, 1\}} \frac{1}{n} \sum_{i=1}^n I(X_i \in G_l, X_i \in R) \mu_t(X_i) \\ & + I(X_i \in G_l, X_i \notin R) \min\{\mu_1(X_i), \mu_{-1}(X_i)\} \\ & - \xi \left(\frac{1}{n} \sum_{i=1}^n I(X_i \in G_l, X_i \in R) \right) - \eta(2 - V(R)), \end{aligned}$$

subject to $n^{-1} I(X_i \in G_l, X_i \in R) > 0$, where $G_1 = \mathcal{X}$, $G_l = \mathcal{X} \setminus \bigcup_{k < l} R_k$ for $l \geq 2$, and $V(R) \in \{0, 1, 2\}$ is the number of parameters used to define clause R .

if $R_l = \mathcal{X}$ **then**

Return $((R_1, t_1), \dots, (R_l, t_l))$

end if

else

 Compute $(R_M, t_M) = \arg \min_{R \in \mathcal{R}, t \in \{-1, 1\}} \frac{1}{n} \sum_{i=1}^n I(X_i \in G_M) \mu_t(X_i) - \eta(2 - V(R))$.

Return $((R_1, t_1), \dots, (R_M, t_M))$

end if

end for

If $l = M$ the solution satisfies $V(R_M) = 0$, and, hence, $R_M = \mathcal{X}$. The estimated decision lists thus have the form

$$\begin{aligned} & \text{If } x \in R_1 \quad \text{then } t_1; \\ & \text{else if } x \in R_2 \quad \text{then } t_2; \\ & \quad \dots \\ & \text{else if } x \in \mathcal{X} \quad \text{then } t_{\hat{M}}, \end{aligned}$$

where $\hat{M} \leq M$. For optimizing over R and t , (Zhang et al., 2018) proposed an algorithm with $\mathcal{O}(n \log n 2d)$ operations. In our implementation, we used a brute-force approach. The hyperparameter grids are given in Table 2.

G.3. Tree-Based Policy Learner

For our tree-based baseline, we make use of the freely available python package EconML (Battocchi et al., 2019). Therein, the idea is to first derive doubly robust estimates of the counterfactual outcomes

$$Y_{i,t}^{\text{DR}} = \mu_t(X_i) + \frac{Y_i - \mu_t(X_i)}{e_t(X_i)} 1\{T_i = t\} \quad (88)$$

and then optimize the policy value in the form

$$\sum_{i=1}^n \left(\frac{1}{2} \pi(X_i) + \frac{1}{2} \right) (Y_{i,1}^{\text{DR}} - Y_{i,-1}^{\text{DR}}). \quad (89)$$

For this baseline, the parameter M describes the maximal depth of the tree.

G.4. Hyperparameters

For all baselines, we use 80% of the data for training and 20% for validation. The hyperparameters are given in Table 2.

Table 2. Hyperparameter Grids

Model	Hyperparameters	Tuning range
Linear eligibility score	regularization parameter ρ	$\{10^{-k} : k \in \{0, \dots, 9\}\}$
Decision list	hyperparameter ξ	$\{0.1, 0.5, 1.0\}$
	hyperparameter η	$\{0.01, 0.1, 1.0\}$
Tree-based policy learner	minimum samples leaf	$\{1, 3, 5\}$
	minimum samples split	$\{5, 10, 20\}$
Deep neural network	initial learning rate	$\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$
	batch size	$\{128, \text{full}\}$
	momentum parameter β_1	$\{0.9, 0.99\}$
	regularization parameter ρ	$\{10^{-2}, 10^{-3}, 10^{-4}\}$

G.5. Scaling

For our baselines, we rescale and shift ψ_i via the formula $\hat{\psi}_i = \frac{\psi_i - \mu}{\sigma}$, where μ is the mean value of ψ and σ the standard deviation. Note that

$$\min_{\pi \in \Pi} \frac{1}{n} \sum_{i=1}^n \hat{\psi}_i I(T_i \neq \pi(X_i)) \quad (90)$$

is equivalent to (EBCP).

H. Illustration of IOPL

H.1. Illustration of Universal Approximation Theorem

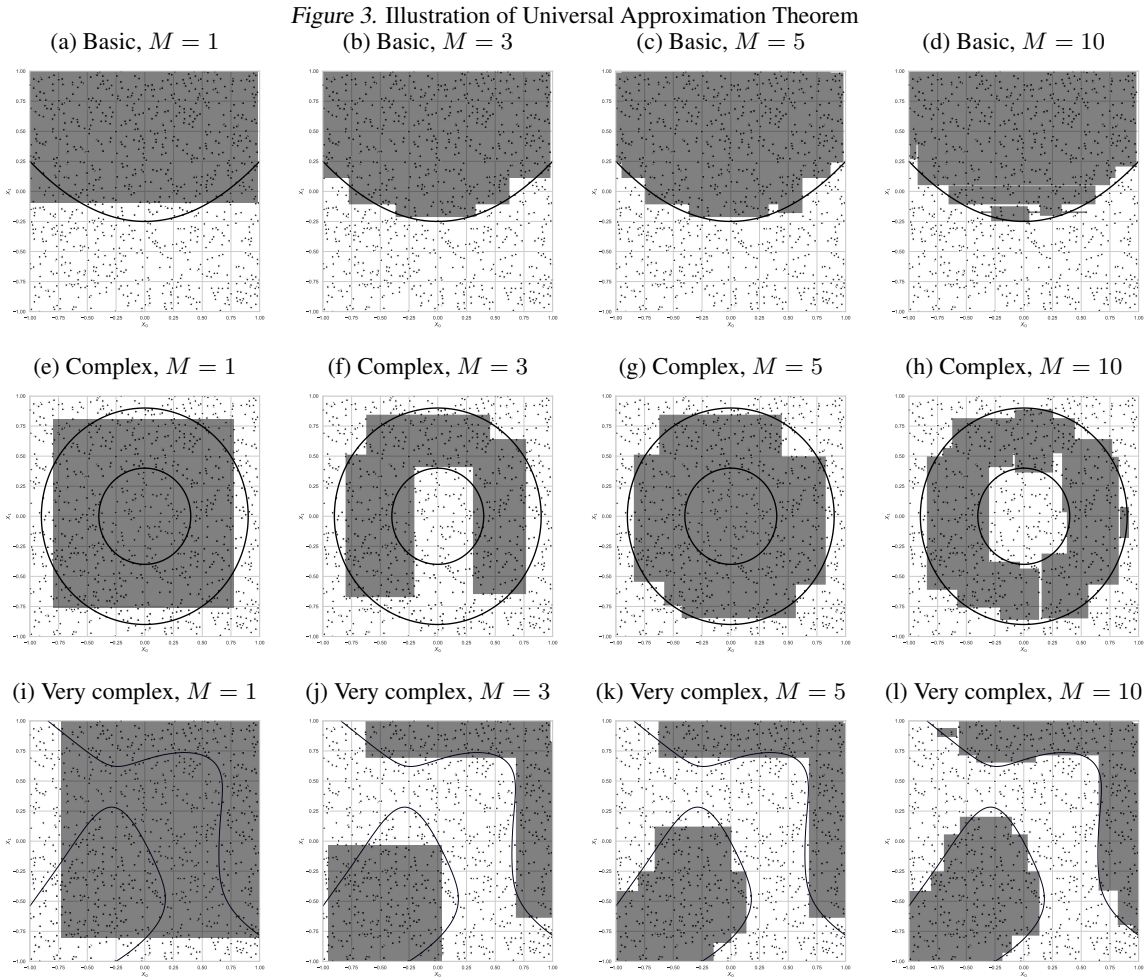
The definitions of the function f for each of the three scenarios are

(1) Basic: $f(X, T) = -2T(X_1 - \frac{1}{2}X_0^2 + \frac{1}{4})$

(2) Complex: $f(X, T) = -2T\left(\left(\frac{4}{10}\right)^2 - X_0^2 - X_1^2\right)\left(X_0^2 + X_1^2 - \left(\frac{9}{10}\right)^2\right)$

(3) Very complex: $f(X, T) = -T\left(3(1-X_0)^2 e^{-X_0^2} - (X_1+1)^2 - 10\left(X_0\frac{1}{5} - X_0^3 - X_1^5\right)e^{-X_0^2 - X_1^2} - \frac{1}{3}e^{-(X_0+1)^2 - X_1^2} - 1\right)$

Figure 3 visualizes the true decision boundaries of the outcome functions in Section 5.1 for all three scenarios. For smaller numbers of hyperboxes, i. e. smaller M , one observes a coarse covering of the decision region, while an increasing number of hyperboxes yields a finer approximation of the decision region. This goes in line with our theoretical findings in Section 4.3.1.



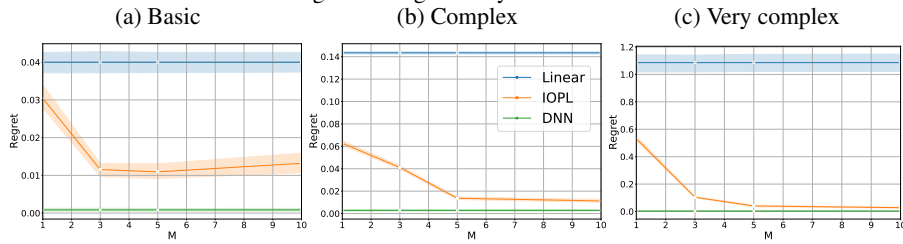
Notes. For demonstration, we used the results from the illustration setting in Section 5.1 (seed=0) with the DR method. The figure shows the hyperboxes returned from IOPL in gray together with the true decision boundaries (black lines). From left to right, one can observe how the parameter M , which controls the number of hyperboxes used in the policy, effects the fit. For $M = 10$, the hyperboxes almost follow the true decision region.

H.2. Performance

In this section, we report the regret as defined in Section 5 for our illustration setting. We sample 10,000 i.i.d. observations according to the three given data-generating processes and use $n = 1,000$ for training. Evidently, DR performs best as can

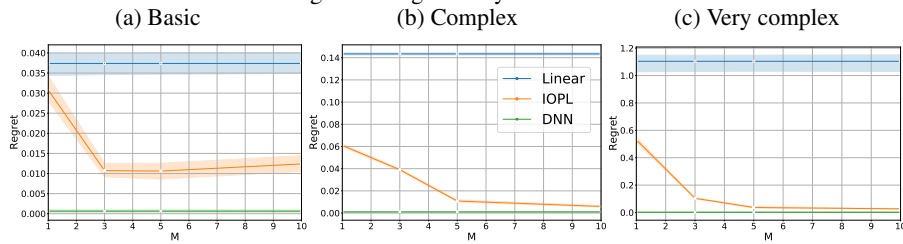
be seen in Figure 4. In the basic scenario, we observe that the linear baseline already performs sufficiently well. Furthermore, we see that the simple form of the decision region in the basic scenario leads to over-fitting for IOPL, which can be seen from the stagnation of the regret for $M > 3$. This goes in line with the observations in Figure 3c and 3d.

Figure 4. Regret analysis with DR



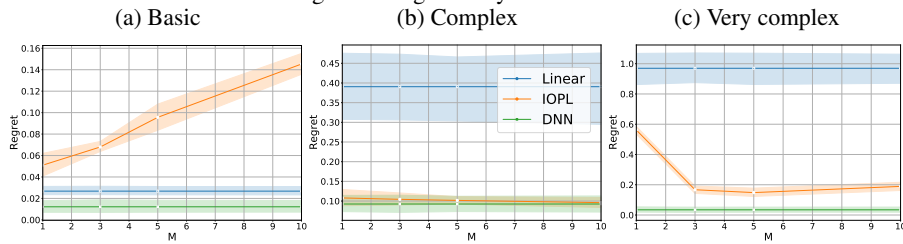
Notes. We set the method to DR. We sample 10 different datasets using the described data-generating processes, which are used for training IOPL and all baselines. For IOPL, we vary the parameter M in $\{1, 3, 5, 10\}$. The shaded areas correspond to the 95% confidence intervals around the mean.

Figure 5. Regret analysis with DM



Notes. We set the method to DM. We sample 10 different datasets using the described data-generating processes. All baselines and IOPL are trained on all 10 datasets. For IOPL, we vary the parameter M in $\{1, 3, 5, 10\}$. The shaded areas correspond to the 95% confidence intervals around the mean.

Figure 6. Regret analysis with IPS



Notes. We set the method to IPS. We sample 10 different datasets using the described data-generating processes. All baselines and IOPL are trained on all 10 datasets. For IOPL, we vary the parameter M in $\{1, 3, 5, 10\}$. The shaded areas correspond to the 95% confidence intervals around the mean.

I. Additional Experiments

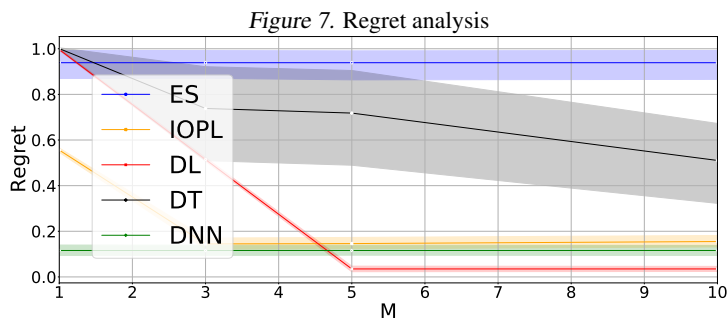
I.1. Regret Analysis in a Simple Setting

In this section we show how IOPL and the baselines perform in a simpler setting. That is, we study a setting in which the optimal policy π^* can be represented as a decision list and a simple decision tree. To do so, we generate again 4-dimensional patient covariates $X_0, X_1, X_2, X_3 \sim U[-1, 1]$, a treatment assignment which is independent of X with $\mathbb{P}(T = 1) = 1/2$, and an outcome $Y \mid X, T \sim \mathcal{N}(m(X, T), \sigma^2)$ with $\sigma = 0.1$. This follows exactly the setting in the regret analysis of Section 5. However, we now set the mean to $m(X, T) = \max(X_2 + X_3, 0) + 1/2 T \text{sign}(X_0 X_1)$. That is, we replace the term $\prod_{i=0}^3 X_i$ by $X_0 X_1$. In this way, it is possible to represent the optimal policy

$$\pi^*(X) = \begin{cases} 1 & \text{if } \text{sign}(X_0) \neq \text{sign}(X_1), \\ -1 & \text{else,} \end{cases} \quad (91)$$

via a decision list and a simple decision tree.

The results of our experiments are summarized in Figure 7. Evidently, now also the baselines are able to approximate the optimal policy and, hence, yield lower regrets. IOPL still outperforms the interpretable baselines with the exception of DL. This is due to the fact that Algorithm 2 is implemented as a brute-force approach and hence yields a global optimum in the training task, while IOPL is stopped after $l = 50$ iterations.



Notes. We sample 10 different datasets using the described data-generating processes. For IOPL, DL, and DT, we vary the parameter M in $\{1, 3, 5, 10\}$. The shaded areas are the 95% confidence intervals.

I.2. Comparison to Optimal Prescriptive Trees (OPTs)

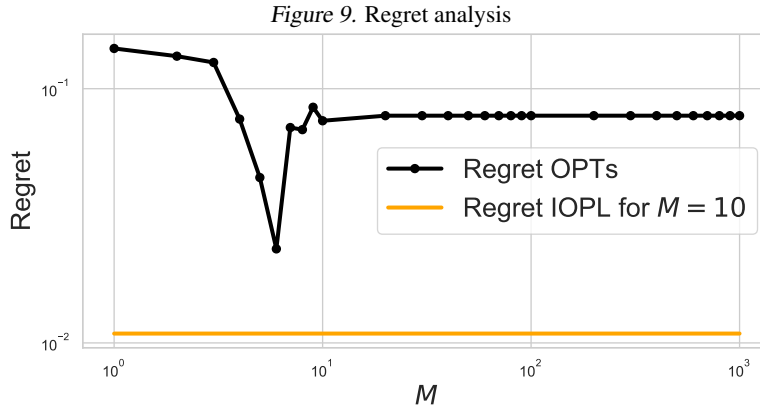
In theory, decision trees are also universal approximators and, hence, are sufficiently rich (see Theorem 4.1 in [Stein & Shakarchi \(2009\)](#) and use that simple functions can be approximated by decision trees). Thus, the lack of theoretical approximation guarantees stems from the fact that heuristic greedy algorithms are used to train decision trees, e. g., CART. That is, decision trees trained via such heuristics cannot leverage their full potential. As a remedy, [Bertsimas et al. \(2019\)](#) recently proposed an algorithm to train so-called optimal prescriptive trees (OPTs). Therein, the authors are building the complete tree at once and, hence, refrain from common heuristic approaches to train decision trees. As a result, decision trees trained with the proposed algorithm become more expressive. A Python implementation of the algorithm is provided in the package [Interpretable AI \(2022\)](#). In the following, we demonstrate that IOPL is still superior to OPTs for two reasons: (i) IOPL is less prone to overfitting. (ii) To ensure that a decision tree has the same approximation quality as a policy from our policy class, one requires very deep trees, which renders them unintelligible.

We demonstrate the first point empirically by fitting OPTs to the data generated in the second scenario in our 2-dimensional simulation study (seed = 0, hyperparameters are given in Table 3) and comparing it to IOPL. We note that OPT is not based on standard off-policy learning techniques, e. g., doubly robust methods. It rather estimates the counterfactual outcomes and the policy value at once via a loss function that is given by a convex combination of the two objectives. Hence, we cannot input the exact nuisance functions in OPT. As a remedy, we also refrain from using the exact nuisance functions in IOPL and rather estimate them from data using scikit-learn. We use a random forest for μ_t and logistic regression for e_t . The hyperparameters for the random forest are given in Table 3. For the logistic regression, we used the standard settings in

scikit-learn. The results are given in Figures 8 and 9. Evidently, OPT is overfitting and cannot benefit from deeper trees, while IOPL yields superior approximations of the true decision region.

Table 3. Hyperparameter Grids

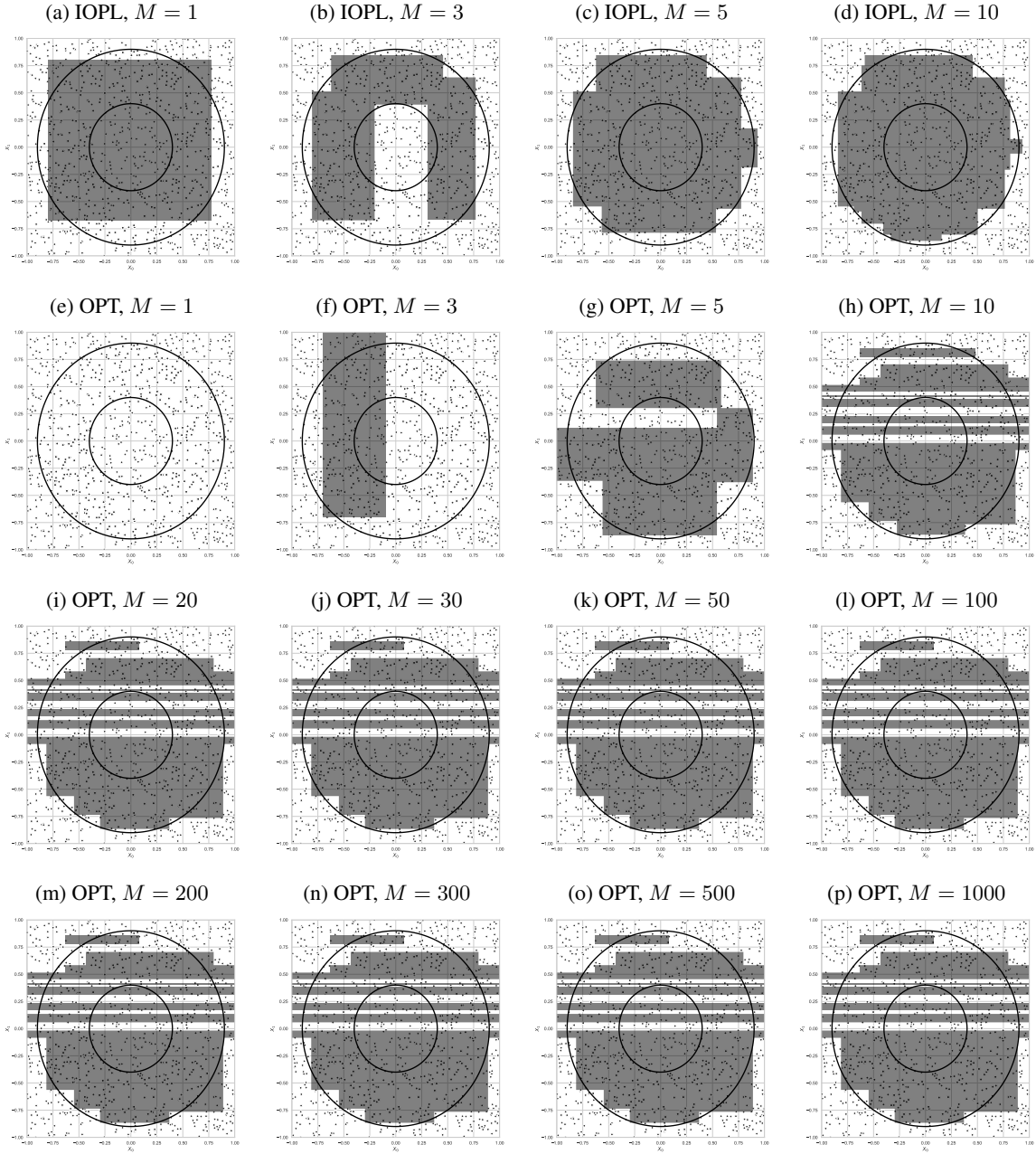
Model	Hyperparameters	Tuning range
Random Forest	minimum number of points in leaf	{1, 3, 5, 10}
	number of trees	{5, 10, 50, 100, 300}
	maximal depth	{5, 10, 50, 100, None}
OPTs	prescription factor γ	{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}
	complexity parameter α	tuned via autotuning procedure in Interpretable AI (2022)
	minimum number of points in leaf	{5, 10, 15}
	maximal depth M	{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, ... 20, 30, 40, 50, 60, 70, 80, 90, 100, ... 200, 300, 400, 500, 600, 700, 800, 900, 1000}



Notes. Regrets of OPT for the above described experiments. M is varied *only* for OPT according to the hyperparameter grid given in Table 3. The baseline regret from IOPL (with $M = 10$) is given in orange.

For the second point, we can think of a worst-case depth that a decision tree has to have, to ensure that a policy in Π_H^M can be represented as a tree. One can easily check that writing a single hyperbox clause as a decision tree requires – in general – a depth of $2d$, where d is the dimension of the covariate space. For every hyperbox that is added, an additional depth of $2d$ is required, as a point can always lie outside the box in just one dimension. That is, to write a policy in Π_H^M as a decision tree, a depth of – in general – $2dM$ is required. For our real-world setting that would result in a worst-case depth of $2 \cdot 12 \cdot 5 = 120$ to represent a policy with $M = 5$. That is, although the same information is represented, our policy class allows for a more compact representation, see, e. g., our policy in Figure 1.

Figure 8. Approximations of Optimal Prescriptive Trees and IOPL



Notes. We used the data generated in the second scenario from the illustration setting in Section 5.1 (seed = 0). The figure shows the decision regions returned from IOPL and OPT in gray together with the true decision boundaries (black lines).

J. Details on Regret Analysis

For the regret analysis, we sample 10,000 observations from the described data-generating process, cf. Equation (45) and (47) in (Athey & Wager, 2021) and use $n = 250$ for training, cf. (Zhang et al., 2018). We train IOPL and all baselines according to Appendix G. The hyperparameter grids for all baselines are given in Table 2.

K. Details on ACTG 175 Study

The ACTG 175 study assigned four treatments randomly to 2,139 subjects with human immunodeficiency virus (HIV) type 1, whose CD4 counts were 200–500 cells/mm³. The four treatments that were compared are the zidovudine (ZDV) monotherapy, the didanosine (ddI) monotherapy, the ZDV combined with ddI, and the ZDV combined with zalcitabine (ZAL).

There are 5 continuous covariates: age (years), weight (kg), CD4 count (cells/mm³) at baseline, Karnofsky score (scale of 0-100), CD8 count (cells/mm³) at baseline. They are scaled before further analysis. In addition, there are 7 binary variables: gender (1 = male, 0 = female), homosexual activity (1 = yes, 0 = no), race (1 = nonwhite, 0 = white), history of intravenous drug use (1 = yes, 0 = no), symptomatic status (1 = symptomatic, 0 = asymptomatic), antiretroviral history (1 = experienced, 0 = naive), and hemophilia (1 = yes, 0 = no). We chose the increase in the CD4 cell count after 20 weeks to be the continuous response.

L. Empirical Policy Values in Experiments with Real-World Clinical Data

As we use real clinical data, we do not have access to the true policy value, which impedes a regret analysis in this setting. For the sake of completeness, we report the empirical policy values in Table 4. Due to the computational complexity of Algorithm 2 for the computation of decision lists, we skipped DL in Table 4.

Table 4. Empirical policy values for experiments with real-world clinical data

Algorithm	Empirical Policy Value		
	$M = 1$	$M = 3$	$M = 5$
IOPL	-0.0353	-0.0433	-0.0540
DT	-0.0326	-0.0326	-0.0349
ES		-0.0021	
DNN		-0.0340	

Notes. Empirical policy values as defined in Equation (EBCP) for IOPL and baselines. Lower is better

M. Questionnaire for User Study

M.1. Questionnaire

We sent the following questionnaire to our participants from clinical practice:

We are currently working on a novel artificial intelligence (AI) method for treatment recommendations in medicine. Can I kindly ask for your help with the following survey (5 min)?

Our research goal

Our goal is to develop a decision support system that recommends treatments to patients. The system will receive characteristics of patients as input (e.g., age, gender, previous diseases). It then returns a recommended treatment as output.

What makes our work unique to earlier research is that our recommendations should be “interpretable” and at the same time “accurate”. We understand the need in medicine to make such treatment recommendations in a manner that is transparent and accountable, as well as, reliable. Specifically, practitioners should understand which treatment is chosen when.

Background

*Many existing tools from artificial intelligence are “black box”. A medical professional would **not** be able to understand which treatment is chosen when. Let’s make an example. Let’s assume Alice, 25 arrives at a medical professional. Let’s further assume she has a medical record with three previous conditions A, B, and C. Then, the artificial intelligence will recommend a treatment based on her gender, age, risk factors, and previous conditions. However, a medical professional would not know which risk factors led to the treatment decision. For example, it would be totally unclear whether age or gender was actually considered (and how). On the contrary, the artificial intelligence would only state treatment 1 or treatment 2 – and that’s it. Aka: only the treatment decision would be presented, without information and without explanation on how to arrive at it. It would thus also not be possible to compare the underlying decision logic with textbook knowledge. As a remedy, interpretable artificial intelligences have been proposed, e. g., eligibility scores, decision lists or decision trees. However, these models have no theoretical guarantees to be expressive enough to capture complicated dependencies between patient characteristics and treatments.*

Example setting

In the following, we would like to assess how interpretable our artificial intelligence is. More specifically, we would like to compare our artificial intelligence against two existing interpretable artificial intelligences in terms of interpretability: (i) linear eligibility score and (ii) decision trees. (We give you later examples of how these look like). However, the existing AI systems come without theoretical guarantees for “accuracy”, so one cannot ensure that these are expressive enough to capture also complicated dependencies (e. g., rare comorbidities).

*For this, we decided upon the following example setting with actual clinical data. In particular, we draw upon the AIDS Clinical Trial Group (ACTG) study 175. The ACTG 175 study assigned two treatments randomly to 1,056 subjects with human immunodeficiency virus (HIV) type 1, whose CD4 counts were 200-500 cells/mm³. The two treatments are: (1) zidovudine (ZDV) monotherapy combined with zalcitabine (ZAL), denoted by **ZDV-ZAL**. (2) zidovudine monotherapy only, in the following denoted by **ZDV**.*

All AI systems will consider the following patient characteristics:

- Age (years)
- Weight (kg)
- Gender (male, female)
- Homosexual activity (yes, no)
- Race (white, non-white)
- History of intravenous drug use (yes, no)
- Symptomatic Status (symptomatic, asymptomatic)
- Antiretroviral history (experienced, naïve)

Interpretable Off-Policy Learning via Hyperbox Search

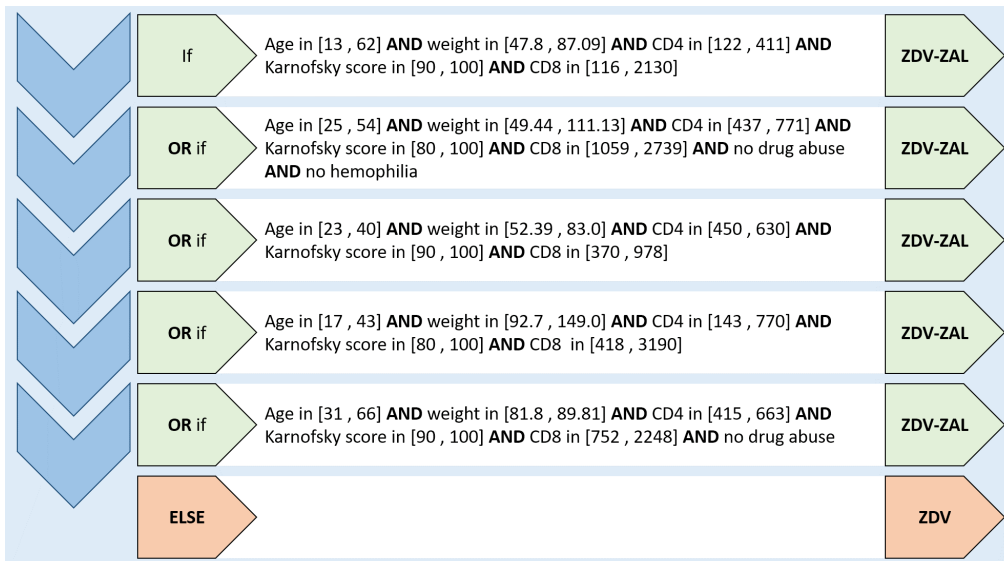
- Hemophilia (yes, no)
- Karnofsky Score (scale of 0–100)
- CD4 count at baseline, i.e., before treatment (cells/mm³)
- CD8 count at baseline, i.e., before treatment (cells/mm³)

SYSTEM 1: Our ‘new’ artificial intelligence

In our ‘new’ artificial intelligence, the treatment recommendation is presented in a simple flow chart. You simply read it from top to bottom. For each “if”, you check whether the characteristics of a patient match the stated condition. For instance, in the first line, you first check whether you treat a patient between 13 and 62 years, and with a weight between 47.8 and 87.09 kg, etc. The recommended treatment – ZDV-ZAL or ZDV – is stated on the right-hand side.

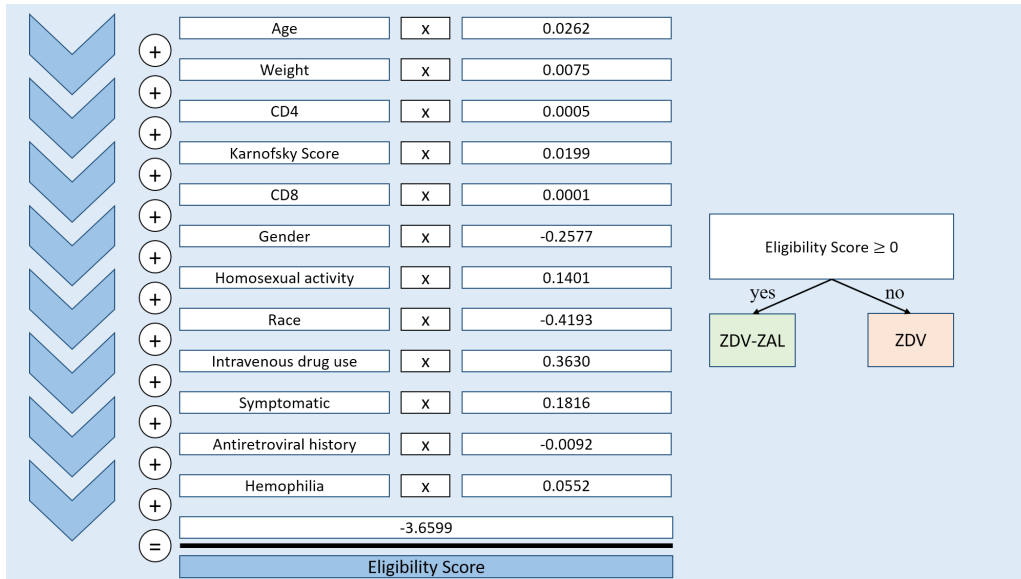
A medical practitioner would thus check which of the “if” conditions is fulfilled, and then pick the corresponding treatment. If one doesn’t fit, just check the next. If none fits, you follow the treatment suggested in the else clause.

Let’s make an illustrative example. So, a patient at the age of 70 would receive ZDV because none of the green condition would apply. A patient at the age of 20, with a weight of 50, a CD4 count at 122, a Karnofsky score at 90, and a CD8 count at 200 would match the first clause and receive ZDV-ZAL.



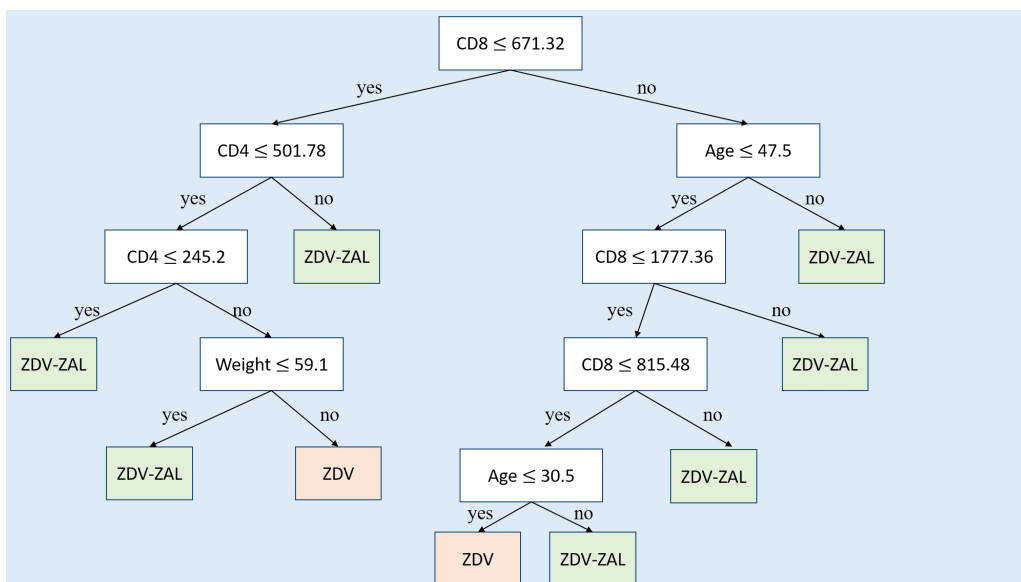
SYSTEM 2: Linear eligibility score

For this artificial intelligence, the treatment recommendation must be manually calculated via a linear mathematical rule. Here, each patient characteristic is first multiplied with a weight (e.g., you multiply patient age with 0.0262, you multiply the patient weight with 0.0075). Afterward, the values are added. The resulting number is called “eligibility score”. If the score is greater or equal to zero, the treatment ZDV-ZAL is chosen. If the score is smaller than zero, treatment ZDV is chosen.



SYSTEM 3: Decision tree

Here the treatment recommendation follows a simple tree structure. You simply check the conditions from top to bottom and follow the tree in order of the corresponding answers: for each white box, you check what the correct answer is. Then you move left if the answer is ‘yes’ and right if the answer is ‘no’. If you arrive at a leaf node, you read off the suggested treatment.



How to fill out the survey

Please find our four survey questions below. For simplicity, we would ask you to just return them in an email.

1. How interpretable do you find the logic in our ‘new’ artificial intelligence? Please rate it in a range from 0 to 10, where 0 is “black box” and 10 “fully transparent”. A 5 would be neutral.
2. How interpretable do you find the logic in linear eligibility scores? Please rate it in a range from 0 to 10, where 0 is “black box” and 10 “fully transparent”. A 5 would be neutral.
3. How interpretable do you find the logic in decision trees? Please rate it in a range from 0 to 10, where 0 is “black box” and 10 “fully transparent”. A 5 would be neutral.
4. Would you consider using treatment recommendations when presented through our ‘new’ artificial intelligence? Yes or no?

M.2. Evaluation

We received the following answers; see Table 5.

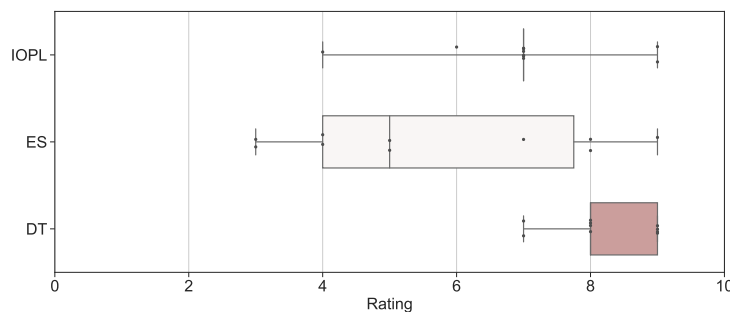
Table 5. Evaluation of Survey

No.	Institution	Job Description	Country	Q 1	Q 2	Q 3	Q 4
1	University Hospital	Medical Doctor	Germany	7	3	8	yes
2	Company	Pharmacist	Sweden	7	4	9	yes
3	Hospital	Clinical Pharmacist	Sweden	9	5	7	yes
4	Private Doctor’s Office	Medical Doctor	Germany	4	8	7	yes
5	University Hospital	Medical Doctor	Germany	7	5	8	yes
6	University Hospital	Medical Doctor	Switzerland	7	7	9	yes
7	Hospital	Medical Doctor	Germany	7	3	8	yes
8	University Hospital	Medical Doctor	Switzerland	6	4	9	no
9	University Hospital	Medical Doctor	Switzerland	7	9	8	yes
10	University Hospital	Radiology specialist	Germany	9	8	9	yes

M.3. Results

The ratings are visualized in Figure 10. Our results demonstrate that, in terms of interpretability, IOPL is largely on par with our baselines with a slight preference for decision trees.

Figure 10. Survey results



N. Computational Considerations in Practice

We note that our definition of policies in Equation (3) is symmetric in the labels $\{1, -1\}$. That is, one can easily change the signs of the training labels and learn a representation of the *no-treatment* decision region. All of our theoretical derivations are of course still valid. In practice, it might thus be useful to compute both representations, as one of them might lead to a more compact policy. That is, in one representation, one might need smaller values of M for the same predictive performance as in the other representation.

O. Limitations

The computational complexity of the pricing problem limits our approach to mid-sized datasets ($n \approx 1,000$ observations) with a moderate dimensionality ($d < 15$). Both of these assumptions are typically fulfilled in clinical practice, as for instance, in the ACTG 175 study used in this study ($n = 1,056$, $d = 12$).

The aim of this paper is to provide interpretable treatment decisions for clinical practice. We think that circumventing the black-box nature of state-of-the-art approaches in off-policy learning, while at the same time yielding policies with low regrets, is crucial to achieve this goal. At the same time, we want to emphasize that medical decisions should ultimately always be made by medical experts. That is, medical experts should always be informed about the risks associated with data-driven algorithms.