UNIVERSITY OF COPENHAGEN
FACULTY OF HEALTH AND MEDICAL SCIENCES

PhD thesis

Statistical policy learning with industrial applications

Andreas Nordland

Advisor: Klaus Holst, Torben Martinussen

Submitted: 18. September 2023

This thesis has been submitted to the Graduate School of the Faculty of Health and Medical Sciences, University of Copenhagen

Preface

This industrial PhD thesis is the result of a cooperation between the Section of Biostatistics at the University of Copenhagen and the Maersk Research Team. Formulating the project together and now seeing the final result has been highly rewarding. Personally, it has also been a privilege to explore a field within statistics with such universal applications that will continue to influence both academia and the industry.

I sincerely thank my supervisors for always being supportive and understanding during tough times.

The project is supported by Innovation Fund Denmark [1].

[1] grant number 8053-00096B

Summary

Transforming Maersk into a data-driven company hinges on successfully leveraging historical data to initiate decision policies wherever applicable. Estimating optimal policies is a causal problem that promotes new ways of collecting, documenting, and generating data. Policy learning is a vast research field across many disciplines. However, the recent development of nonparametric and doubly robust policy learning techniques in statistics and economics has yet to find applications in logistics and other industries. We see massive potential for these assumption-lean techniques to leverage the vast amounts of historical data in Maersk and initiate a data-driven operation and business culture.

The primary objective of this project is to ease the practical application of the latest theoretical developments. A key contribution is the comprehensive R package `polle`, which unifies existing policy learning methods, introduces new functionality, and ensures consistent policy evaluation.

To illustrate the usefulness of this implementation, we present a novel application aimed at optimizing maintenance and repair policies to maximize the long-term utility of reefers. A central challenge in this application is to address practical positivity violations arising from limited variation in the decision process. We advocate for a solution involving an action probability threshold restriction, resulting in an estimate for the optimal realistic work order policy. Our findings indicate a significant gain in value, amounting to an estimated \$7.5 million increase in annual profits.

For cases involving extended follow-up periods, obtaining an early indication of the effectiveness of an action or treatment using a post-randomization response indicator is highly valuable. The final contribution of this project focuses on studying the treatment effect among responders, defined as a principal stratum. For a survival analysis setup, we make novel contributions to dealing with right censoring and construct a nonparametric efficient estimator for the target parameter. This target parameter is applicable for subgroup analysis or for designing optimal treatment-switching policies when combined with policy learning techniques.

Dansk resumé

Realiseringen af Mærsk's ambition om at blive en datadreven virksomhed er betinget af, at man successfuldt kan omsætte historisk data til forbedrede beslutningsregler i alle områder af forretningen. At estimere en optimal beslutningsregel er et kausalt problem, der fordrer ny måder at samle, dokumentere og generere data. Optimering af beslutningsregler er et stort forskningsfelt på tværs af mange discipliner. Dog har nylige fremskridt inden for ikke-parametrisk og dobbelt robust estimation af den optimale beslutningsregel endnu ikke fundet anvendelse inden for logistik og andre industrier. Vi ser et enormt potentiale af disse teknikker, som beror på et minimum af antagelser. Maersk kan således udnytte sin kæmpe mængde historisk data til at påbegynde en datadreven arbejdskultur i alle dele af virksomheden.

Det primære formål med dette projekt er at lette den praktiske anvendelse af de senest udviklede metoder. Et centralt bidrag er den omfattende R pakke `polle`, som forener eksisterende metoder, introducerer nye funktioner og sikrer konsistent evaluering af den estimerede beslutningsregel.

For at illustrere nyttigheden af vores implementering præsenterer vi en original anvendelse med henblik på at optimere den langsigtede nytte ved vedligehold og reparationer af kølecontainere. En central udfordring ved denne anvendelse er at håndtere praktiske krænkelser af positivitetsantagelsen, der opstår pga. begrænset variation i den eksisterende beslutningsproces. Vi foreslår at benytte en grænseværdi for sandsynligheden af den anbefalede beslutning således, at vi kun optimerer over sættet af realistiske beslutningsregler. Analysen angiver en signifikant gevinst på årligt \$7.5 millioner.

I sager med lange opfølgningsperioder er det være værdigfuldt at få en tidlig indikation, om beslutningen eller behandlingen har haft den ønskede virkning. Det sidste bidrag af dette projekt omhandler identifikation og estimation af behandlingseffekten blandt de responderende enheder defineret som et postrandomiseringsstratum. I forbindelse med overlevelsesanalyser introducerer vi en ny måde at håndtere højrecensurering og konstruerer samtidigt en effektiv ikke-parametrisk estimator. Metoden er relevant for undergruppeanalyser og for at designe individualiserede beslutningsregler på baggrund responsindikatorerne.

Contents

Chapter 1

Introduction

As pointed out in an essay by Susan Athey in Science [2], there are several gaps between making predictions and making decisions. In Maersk and across logistics, investments in technology have facilitated the use of machine learning prediction methods. As a result, an increasing number of models are being put into production to continuously produce demand forecasts, estimate customer churn, predict asset component failures, etc. These predictions are then utilized to allocate and reposition containers, initiate customer care actions, and conduct inspections. While significant attention has been given to the development of data infrastructure, model deployment, and computational resources, there remains a lack of understanding of the assumptions needed to make meaningful data-driven decisions.

For example, freight rates are positively correlated with demand for freight due to limited supply. Thus, a naive interpretation of a simple demand forecast with the freight rate as input would suggest that higher rates increase demand. However, in reality, actively raising prices would likely decrease demand in a given situation.

Similarly, customer churn models can successfully predict the churn propensity of customers based on their booking history. However, it is not clear how to effectively use such models to prioritize customer care initiatives [1]. Should Maersk spend resources targeting customers who are most likely to churn? It would probably be more sensible to target customers who are likely to churn and who are likely to respond positively to being contacted by Maersk.

All of these considerations are even more complicated for sequential decision problems. It is unclear how to adjust for confounding in these cases using standard prediction models. With the vast amount of historical data available at Maersk, documenting decisions across the entire organization, the question is:

How can we effectively utilize existing historical data at Maersk to improve decision policies?

Online methods in reinforcement learning are designed to optimize sequential decision processes. However, these methods rely on a model of the environment or require iterative experiments. Conducting experiments can be extremely costly and impractical, especially for extended follow-up periods. Offline reinforcement learning addresses this problem for Markov decision problems. Similarly, statistical policy learning has experienced significant development in economics and health sciences, where it is employed to estimate optimal dynamic treatment regimes. This industrial PhD project was scoped based on these advancements, mainly focusing on doubly robust policy learning methods. From our perspective, these assumption-lean methods hold substantial potential for the industry, especially for companies like Maersk.

Firstly, these methods are tailored to have a causal interpretation while relying on a minimal set of structural assumptions. Secondly, the methods are focused on constructing non-parametric efficient estimators, enabling the use of flexible machine learning nuisance models without sacrificing valid performance guarantees or inference for the estimated value gain when implementing the learned policy. Furthermore, reinforcement learning formulations have often lacked an understanding of the impact of nuisance model estimation, which is an area where these doubly robust policy learning methods can offer valuable insights. By leveraging these advantages, we believe that applying and adapting these methods to Maersk's historical data will help transform the company into being truly data-driven in many aspects of operations and business. Still, we see some clear challenges that need to be solved before these methods can be applied on a large scale in a company like Maersk.

Research Objective 1: Software implementation

In our view, there exists a considerable lack of software implementations focusing on the latest developments within statistical policy learning that can effectively evaluate and compare a range of different policy learners while providing a flexible setup for specifying each of the model components. The first major contribution of the project is thus a comprehensive software implementation resulting in the R package `polle` and the associated Paper A. The package is built around three central components: a policy data object, a policy learner setup, and a policy evaluation functionality. The policy data object allows the user to easily structure the data for decision problems with a fixed or stochastic number of stages. The policy learner setup provides a unifying framework by combining already available methods from other packages with new learners not readily available elsewhere. It is also easy for the user to define static or dynamic policies and create new policy learning methods. The performance of each policy or policy learner can then be evaluated using the policy evaluation functionality with easy specification of the required nuisance models and cross-fitting setup.

Research Objective 2: Novel industrial application

Maersk owns an extensive portfolio of assets, including ships, containers, and terminals. Asset management is thus an integral part of the business. Equipment maintenance is classically formulated as a Markov decision problem, as illustrated by examples like the bus engine replacement problem [39]. Thus, at the beginning of the project, the repair and maintenance of containers, particularly refrigerated containers was identified as an excellent application of statistical policy learning. And with the project anchored in the Section of Biostatistics at the University of Copenhagen, it was easy to draw an analogy to statistical policy learning within personalized medicine. Each refrigerated container is a patient with an expected lifetime of 16-20 years. Whenever the box or the refrigeration unit is damaged or requires maintenance, a work order from the designated repair shop needs to be approved. Like in quality of life research, we do not solely want to optimize the expected lifetime, but rather a utility measure that balances costs with how much the container has been used. Paper B documents this novel application of statistical policy learning within asset management.

Research Objective 3: Policy learning under positivity violations

One particular challenge that we faced in the above application was a lack of treatment variation due to existing enforced guidelines. This leads to positivity violations, which is one of the key causal assumptions needed to identify the optimal policy. Our proposed solution to this problem is to restrict the class of policies to actions deemed realistic at a given probability threshold. To our knowledge, the polle package is the only available software that includes this functionality. The importance of protecting doubly robust policy learners against positivity violations is highlighted in a novel simulation study that mimics the lack of treatment support experienced in the application.

Research Objective 4: Treatment effect among responders

For actions or treatments where the outcome is observed after a long duration of time, it is highly valuable if we can get an early indication using a biomarker, for example, of whether the patient responds to the treatment. This would allow us to switch treatments quickly and optimize the expected outcome. In a business context, this is also highly relevant. For example, we can investigate the effectiveness of various marketing campaigns. What is the effect of each campaign among the customers who were actually exposed to it? The final Paper C studies the average treatment effect among responders in a survival setup. Conditioning on a post-randomization response variable will generally not yield a causal interpretation. Thus, we condition on the principal stratum of treatment responders instead. The key assumption allowing us to identify the treatment effect among responders is that non-responders have

no effect of the treatment. This strong but easily interpretable assumption will need to be justified from case to case. The results from these types of analyses can inspire the design of sequentially randomized trials verifying the long-term advantage of treatment-switching regimens. Contributions of Paper C include novel considerations for conditionally independent right censoring and the construction of the associated non-parametric efficient one-step estimate for the treatment effect among responders.

In non-parametric estimation theory, the concept of efficient influence functions and the associated estimators is key. All of the work in this thesis heavily depends on this methodology. For completeness, we briefly introduce efficient non-parametric estimation in Chapter 2. Chapter 3 provides a comprehensive introduction to policy learning, covering a wide range of details on identification and performance measures not covered in Paper A and Paper B, and with an emphasis on our contributions. Chapter 4 presents the background and motivation for optimizing the repair and maintenance policy for refrigerated containers as covered in Paper B. Chapter 5 introduces the concept of treatment effects among responders as studied in Paper C and discusses its connection to policy learning. Finally, Chapter 6 discusses our research objectives, possible extensions of our implementation and methods, along with potential areas for future research.

Chapter 2

Efficient non-parametric estimation

This chapter serves as a short introduction to efficient non-parametric estimation theory. It is a reflection (or perhaps a projection) of the rich theory around influence functions. For more comprehensive coverage, we recommend the reader to explore reviews by [8, 55, 53]. Additionally, we will highlight recent intuitive reviews by [15, 17].

The workhorse of non-parametric estimation is the concept of parametric submodels. Let $\mathcal{P}$ be a collection of distributions. For the true distribution $P_0 \in \mathcal{P}$ we consider submodels $P_\epsilon \in \mathcal{P}$, $\epsilon > 0$, for which P_ϵ is differentiable in $\epsilon = 0$ with score s. For simplicity, we assume that P_ϵ is absolutely continuous with respect to a dominating measure ν. We let f_ϵ denote the density of P_ϵ. The score of the submodel is then given by

$$s(x) = \frac{\partial}{\partial \epsilon} \log(f_\epsilon(x))\Big|_{\epsilon=0},$$

where $s(X)$ is an element in the $L_2(P_0)$ space. The set of all parametric submodel scores is called the tangent set and is denoted $T(P_0)$.

In this work, the target parameter will always be defined as a function of the true distribution onto the real line $\Psi(P_0) \in \mathbb{R}$. We say that the target parameter is pathwise differentiable at P_0 if for every score s in the tangent set $T(P_0)$ and for every parametric submodel with score s there exist a continuous linear function $\dot{\Psi}(P_0) : L_2(P_0) \mapsto \mathbb{R}$ such that

$$\frac{\Psi(P_\epsilon) - \Psi(P_0)}{\epsilon} \to \dot{\Psi}(P_0)(s).$$

Let the tangent space $\overline{T}(P_0)$ be the closed linear span of the tangent set. By the Riesz representation theorem, the map $\dot{\Psi}(P_0)$ can be represented by an inner product with a unique function $\psi(P_0) \in \overline{T}(P_0)$. Specifically,

$$\dot{\Psi}(P_0)(s) = \langle \psi(P_0), s \rangle_{L_2(P_0)} = \int \psi(P_0)(X) s(X) dP_0. \tag{2.1}$$

Note that for any element $s^\perp$ in the orthogonal complement to the tangent space, it also holds that

$$\dot{\Psi}(P_0)(s) = \langle \psi(P_0) + s^\perp, s \rangle_{L_2(P_0)}.$$

We denote the linear variety $\psi(P_0) + \overline{T}(P_0)^\perp$ the set of influence functions and $\psi(P_0)$ the efficient influence function. We say that a model is non-parametric if $\overline{T}(P_0) = L_2(P_0)$. This follows from the fact that if we put no restrictions on $\mathcal{P}$, a valid submodel is given by

$$f_\epsilon(x) = (1 + \epsilon s(x)) f(x)$$

for any mean zero bounded score function s. The closed linear space of these scores is the maximal tangent space $L_2(P_0)$. Thus, only a single influence function exists.

For a given parametric submodel, the optimal asymptotic variance is bounded by the Cramer-Rao lower bound:

$$\frac{\frac{\partial}{\partial \epsilon} \Psi(P_\epsilon)|_{\epsilon=0}}{\mathbb{E}[s(X)^2]} = \frac{\langle \psi(P_0), s \rangle_{L_2(P_0)}}{\langle s, s \rangle_{L_2(P_0)}},$$

where $\mathbb{E}[s(X)^2]$ is the Fisher information. Intuitively, the asymptotic lower bound for a semi-parametric estimator must be larger than the asymptotic lower bound for any submodel or, equivalently, any score in the tangent set. Thus, a lower bound for the asymptotic variance of a semi-parametric estimator is given by the variance of the efficient influence function:

$$\sup_{s \in \overline{T}(P_0)} \frac{\langle \psi(P_0), s \rangle_{L_2(P_0)}}{\langle s, s \rangle_{L_2(P_0)}} = \mathbb{E}[\psi(P_0)(X)^2],$$

This result follows by the Cauchy-Schwarz inequality and the fact that $\psi(P_0)(X)$ is an element of $\overline{T}(P_0)$. Let $\hat{\Psi}_n$ denote a regular asymptotically linear estimator of $\Psi(P_0)$ based on a sample of n iid observations. Formally, the estimator is efficient under the model $\mathcal{P}$ if and only if

$$\hat{\Psi}_n - \Psi(P_0) = n^{-1} \sum_{i=1}^{n} \psi(P_0)(X_i) + o_{P_0}(n^{-1/2}). \tag{2.2}$$

Assuming that we know the efficient influence function, how do we construct an estimator that fulfills (2.2)? We will use the notation $P_n V = n^{-1} \sum_{i=1}^n V_i$ for n iid variables $(V_i)_{i \in \{1,\dots,n\}}$ and $PV = \int V dP$. Furthermore, let $\hat{P}_n$ denote an estimator of the model P_0. For any plug-in estimator $\Psi(\hat{P}_n)$ we can write

up the so-called von Mises expansion:

$$\begin{aligned}
&\Psi(\hat{P}_n) - \Psi(P_0) \\
&\quad = P_n\psi(P_0)(X) \\
&\qquad - \; P_n\psi(\hat{P}_n)(X) && (2.3)\\
&\qquad + \; \{P_n - P_0\}\left\{\psi(\hat{P}_n)(X) - \psi(P_0)(X)\right\} && (2.4)\\
&\qquad + \; R(\hat{P}_n, P_0), && (2.5)
\end{aligned}$$

where the second order remainder is given by

$$R(\hat{P}_n, P_0) = P_0\psi(\hat{P}_n)(X) + \left\{\Psi(\hat{P}_n) - \Psi(P_0)\right\}.$$

If the bias remainder (2.3), the empirical process remainder (2.4), and the second order remainder are all $o_{P_0}(n^{-1/2})$, then by the central limit theorem, the estimator is consistent and asymptotically linear, with the variance given by the variance of the efficient influence function.

As described in [54], to prove that the empirical process remainder (2.4) is $o_{P_0}(n^{-1/2})$, it is sufficient to show that $\psi(\hat{P}_n)$ falls into a Donsker class with probability tending to one, and that conditional on $\hat{P}_n$ it holds that

$$\|\psi(\hat{P}_n)(X) - \psi(P_0)(X)\|_{2,P_0} = o_{P_0}(1).$$

However, if $\hat{P}$ is fitted on a separate dataset, such as through cross-fitting, we may lose the Donsker class condition. It may seem complicated to show that the second order remainder term is $o_{P_0}(n^{-1/2})$, but in many cases, the term simplifies considerably as we will also see in the next section. The real problematic term is the bias term. However, we can avoid this term by simply adjusting for the bias. The resulting estimator is the so-called one-step estimator given by

$$\Psi(\hat{P}_n) + P_n\psi(\hat{P}_n)(X).$$

Other estimators, such as estimation equation estimators based on the efficient influence function, may also directly imply that the bias term is zero.

Only a single (efficient) influence function exists for a non-parametric estimator. Thus, given a candidate for the efficient influence function and the associated one-step estimator, if we can show that the remainder terms are $o_{P_0}(n^{-1/2})$, then the estimator will be efficient.

To summarise, our strategy for non-parametric efficient estimation is: 1) Find a candidate for the efficient influence function, for example, by calculating the Gateaux derivative. 2) Construct a cross-fitted one-step estimator or another cross-fitted estimator for which the bias term is zero. 3) Show that the empirical process term and second order remainder term are $o_{P_0}(n^{-1/2})$. 4) Estimate the variance of the estimator via the fitted influence function.

2.1 Gateaux derivaties

The Riesz representation theorem (2.1) provides a direct method for calculating the efficient influence function. However, for non-parametric models, the calculations involved can become unnecessarily complicated.

An alternative approach, advocated by [15, 17, 21], is to calculate the Gateaux derivative of the target parameter using a point mass contamination. This derivative is a good candidate for the efficient influence function. The Gateaux derivative is simply the pathwise derivative for a single specific parametric submodel. For a distribution $\tilde{P}$, the Gateaux derivative of Ψ around P_0 in the direction $\tilde{P}$ exists if

$$\frac{\Psi(P_\epsilon) - \Psi(P_0)}{\epsilon} \to \int \psi(P_0)(x) d(\tilde{P}(x) - P(x))$$

where P_ϵ has density $f_\epsilon(x) = f_0(x) + \epsilon(\tilde{f}(x) - f_0(x))$. If $\Psi(P_0)$ is pathwise differentiable, then the efficient influence will equal the function $\psi(P_0)$. Specifically,

$$\int \psi(P_0)(x) d(\tilde{P}(x) - P(x)) = \int \psi(P_0)(x) \left(\frac{\tilde{f}(x)}{f_0(x)} - 1 \right) dP_0(x),$$

where $\frac{\tilde{f}(x)}{f_0(x)} - 1$ is the score of the submodel. Even if the data is continuous, we will assume it is discrete computationally. This allows us to consider point mass contaminations $\tilde{f}(x) = I_{\tilde{x}}(x)$. Under this submodel and using the fact that the efficient influence function has mean zero, we get that the Gateaux derivative directly equals the efficient influence function evaluated at $\tilde{x}$:

$$\frac{\Psi(P_\epsilon) - \Psi(P_0)}{\epsilon} \to \psi(P_0)(\tilde{x}).$$

An advantage of working with point mass contamination Gateaux derivatives is that the chain rule applies. Thus, we can decompose the Gateaux derivative of a target parameter into standard expressions. For the remainder of this section, we will report some useful results and display some key examples for our work.

For a submodel over the variables X and Y, it is useful to know that the marginal sub-model is given by

$$\begin{aligned} f_\epsilon(x) = \int f_\epsilon(y, x)\, d\nu(y) =& \epsilon I_{\tilde{x}}(x) \int I_{\tilde{y}}\, d\nu(y) + (1 - \epsilon) \int f_0(y, x)\, d\nu(y) \\ =& \epsilon I_{\tilde{x}}(x) + (1 - \epsilon) f_0(x). \end{aligned}$$

This result directly implies that the conditional sub-model is given by

$$f_\epsilon(y \mid x) = \frac{f_\epsilon(y, x)}{f_\epsilon(x)} = \frac{f_0(y, x) + \epsilon \cdot (I_{\tilde{y}}(y) I_{\tilde{x}}(x) - f_0(y, x))}{f_0(x) + \epsilon \cdot (I_{\tilde{x}}(x) - f_0(x))}.$$

Lemma 1. *Assume* $f_0(x) = \mathbb{P}(X = x) > 0$, *then*

$$\begin{aligned}\frac{\partial}{\partial\epsilon} f_\epsilon(y|x)\Big|_{\epsilon=0} &= \frac{I_{\tilde{y}}(y)I_{\tilde{x}}(x) - f_0(y,x)}{f(x)} - \frac{f_0(y,x)[I_{\tilde{x}}(x) + f_0(x)]}{f_0(x)_0^2} \\ &= I_{\tilde{y}}(y)I_{\tilde{x}}(x)f_0(x)^{-1} - I_{\tilde{x}}(x)f_0(y|x)f_0(x)^{-1}\end{aligned}$$

Example 1. For the target parameter $\mathbb{E}(Y \mid X = x)$ the point mass contamination Gateaux derivative is given by

$$\begin{aligned}\psi(P_0)(\tilde{y},\tilde{x}) =& \int yI(\tilde{y} = y)I(\tilde{x} = x)f_0(x)^{-1}\, d\nu(y) \\ &- \int yI(\tilde{x} = x)f_0(y|x)f_0(x)^{-1}\, d\nu(y) \\ =& \frac{I(\tilde{x} = x)}{\mathbb{P}(X = x)}\{\tilde{y} - \mathbb{E}(Y|X = x)\}.\end{aligned}$$

Lemma 2. *For a (Gateaux) differentiable function* $v(P_0)(x)$ *it holds that the point mass contamination Gateaux derivative of* $\mathbb{E}[v(P_0)(X)]$ *is given by*

$$\psi(P_0)(\tilde{x}) = \mathbb{E}\left[v_1(P_0)(X)\right] + v(P_0)(\tilde{x}) - \mathbb{E}[v(P_0)(X)]$$

where $v_1(P_0)(X) = \frac{\partial}{\partial\epsilon} v(P_\epsilon)(X)\big|_{\epsilon=0}$.

Example 2. For variables (H, A, U) assume that $\mathbb{P}(A = a|H) > 0$ for some a. Let the target parameter be given by

$$\Psi(P_0) = \mathbb{E}\left[\mathbb{E}[U \mid A = a, H]\right].$$

Define

$$v(P_0)(H) = \mathbb{E}[U \mid A = a, H].$$

Then

$$\frac{\partial}{\partial\epsilon} v(P_\epsilon)(h)\Big|_{\epsilon=0} = \frac{I(\tilde{a} = a)I(\tilde{h} = h)}{\mathbb{P}(A = a, H = h)}\left\{\tilde{u} - \mathbb{E}[U|A = a, H = \tilde{h}]\right\},$$

and

$$\begin{aligned}&\int \frac{I(\tilde{a} = a)I(\tilde{h} = h)}{\mathbb{P}(A = a, H = h)}\left\{\tilde{u} - \mathbb{E}[U|A = a, H = \tilde{h}]\right\} f_0(h)d\nu(h) \\ =& \frac{I(\tilde{a} = a)}{\mathbb{P}(A = a|H = \tilde{h})}\left\{\tilde{u} - \mathbb{E}[U|A = a, H = \tilde{h}]\right\}.\end{aligned}$$

Thus, the Gateaux derivative is given by

$$\begin{aligned}\psi(P_0)(\tilde{h},\tilde{a},\tilde{u}) =& \frac{I(\tilde{a} = a)}{\mathbb{P}(A = a|H = \tilde{h})}\left\{\tilde{u} - \mathbb{E}[U|A = a, H = \tilde{h}]\right\} \\ &+ \mathbb{E}[U \mid A = a, H = \tilde{h}] - \Psi(P_0).\end{aligned}$$

Chapter 3

Policy learning

Learning better policies from historical data is at the core of this thesis. Unlike online policy learning methods, where it is possible to interact with the environment, such as when playing a game, learning the effects of alternative actions based on historical data requires careful causal considerations. In this regard, a limited amount of available data necessitates efficient learners. Otherwise, we cannot make any performance guarantees or estimate the value of the learned policy. The purpose of this chapter is not to reiterate the content covered in Paper A and Paper B, but rather to expand on key results not included in the manuscripts and highlight our contributions. Additionally, we will introduce new material concerning variable importance measures for the estimated policy.

The policy learning process can be divided into two parts. The first part involves formally defining the optimal policy as a causal parameter and determining the structural assumptions that allow us to identify it from the data. This part is described in detail in Sections 3.1 and 3.2. The second part revolves around estimating the policy and evaluating its performance. Doubly robust policy estimation is introduced in Section 3.3, and efficient performance measures are described in Section 3.4. Finally, material on functional inference for the policy, which is not considered in any of the papers, is included in Section 3.7.

Most of this chapter focuses on a simplified two-stage decision problem. A general K-stage formulation will not add any conceptual understanding but will introduce a lot of cumbersome notation. We represent a single observation as:

$$O = (S_1, A_1, S_2, A_2, U).$$

Here, $S_1 \in \mathcal{S}_1$ and $S_2 \in \mathcal{S}_2$ represent general state variables, $A_1 \in \mathcal{A}_1$ and $A_2 \in \mathcal{A}_2$ are binary action or decision variables, and U denotes the utility outcome. For convenience, define the history variables $H_1 = S_1 \in \mathcal{H}_1$ and $H_2 = (S_1, A_1, S_2) \in \mathcal{H}_2$.

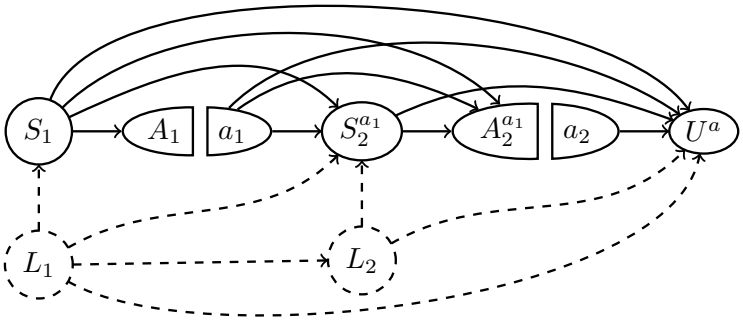


Figure 3.1: Single world intervention graph (SWIG). S_1 and $S_2^{a_1}$ represent the potential state variables, A_1, and $A_2^{a_1}$ represent the potential action variables and U^a represents the potential utility outcome. In this case, the assumption of sequential randomization holds even though L_1 and L_2 are unmeasured variables.

3.1 Optimality

In this section, we will formally define what the optimal policy means in a causal framework. For any static actions $a = (a_1, a_2) \in \mathcal{A} = \{0, 1\}^{\otimes 2}$, we introduce the potential outcomes [43] under action a as follows:

$$O^a = (S_1, a_1, S_2^{a_1}, a_2, U^a).$$

The potential outcome O^a represents the observation we would have observed if, contrary to the fact, we had forced the actions to be $A_1 = a_1$ and $A_2 = a_2$. We will not go into a discussion on the existence of potential outcomes. However, in practical terms, we always think of potential outcomes as the result of an intervention in a structural equation model. This also allows us to visualize interventions in graphs [41]. Figure 3.1 displays an example of a single-world intervention graph. The variables L_1 and L_2 represent unmeasured variables.

Let $V_1 \in \mathcal{V}_1$ be a function of H_1 and let $V_2 \in \mathcal{V}_2$ be a function of H_2. A policy restricted to the input V_1 and (A_1, V_2) is a set of rules $d = (d_1, d_2)$ where $d_1 : \mathcal{V}_1 \mapsto \mathcal{A}_1$ and $d_2 : \mathcal{A}_1 \times \mathcal{V}_2 \mapsto \mathcal{A}_2$. Based on a given policy d, we define the potential utility under d as follows:

$$U^d = \sum_{a_1 \in \mathcal{A}_1, a_2 \in \mathcal{A}_2} U^{a_1, a_2} I\{d_2(a_1, V_2^{a_1}) = a_2\} I\{d_1(V_1) = a_1\}.$$

Hypothetically, we would want to maximize U^d directly for every observation, but unfortunately, we only observe a single version of the world. Instead, we

define the optimal V-restricted policy as follows:

$$d_0 = \arg\max_{d \in \mathcal{D}} \mathbb{E}[U^d].$$

Here, $\mathcal{D}$ is the class of all V-restricted policies. This definition is not very constructive in terms of how we should estimate the optimal policy. However, with a bit more structure on the inputs V_1 and V_2, we can prove the following important theorem. The proof is a corrected version of Theorem 1 in [52].

Theorem 3. *If V_1 is a function of V_2, then the V-optimal policy d_0 is given by*

$$\begin{aligned} B_{0,2}(a_1, v_2) &= \mathbb{E}[U^{a_1,a_2=1} | V_2^{a_1} = v_2] - \mathbb{E}[U^{a_1,a_2=0} | V_2^{a_1} = v_2] \\ d_{0,2}(a_1, v_2) &= I\left\{B_{0,2}(a_1, v_2) > 0\right\} \\ B_{0,1}(v_1) &= \mathbb{E}[U^{a_1=1,d_{0,2}} | V_1 = v_1] - \mathbb{E}[U^{a_1=0,d_{0,2}} | V_1 = v_1] \\ d_{0,1}(v_1) &= I\left\{B_{0,1}(v_1) > 0\right\}. \end{aligned}$$

The above statement is also true if for all a_1 and a_2

$$\mathbb{E}[U^{a_1,a_2} | V_1, V_2^{a_1}] = \mathbb{E}[U^{a_1,a_2} | V_2^{a_1}]. \tag{3.1}$$

Proof. Let $V^a = (V_1, V_2^a)$. For any policy d

$$\begin{aligned} \mathbb{E}[U^d] &= \mathbb{E}\left[\sum_{a_1,a_2} U^{a_1,a_2} I\{d_2(a_1, V_2^{a_1}) = a_2\} I\{d_1(V_1) = a_1\}\right] \\ &= \sum_{a_1} \mathbb{E}\left[\left\{\sum_{a_2} \mathbb{E}\left(U^{a_1,a_2} \middle| V_2^{a_1}\right) I\{d_2(a_1, V_2^{a_1}) = a_2\}\right\} I\{d_1(V_1) = a_1\}\right], \end{aligned}$$

where it is used that V_1 is a function of $V_2^{a_1}$ or that (3.1) holds. For any a_1 the inner sum is maximized in d_2 by $d_{0,2}$, i.e., $\mathbb{E}[U^d] \leq \mathbb{E}[U^{d_1,d_{0,2}}]$. Now,

$$\mathbb{E}[U^{d_1,d_{0,2}}] = \mathbb{E}\left[\sum_{a_1} \mathbb{E}[U^{a_1,d_{0,2}} | V_1] I\{d_1(V_1) = a_1\}\right],$$

which is maximized for $d_1 = d_{0,1}$, i.e., $\mathbb{E}[U^d] \leq \mathbb{E}[U^{d_1,d_{0,2}}] \leq \mathbb{E}[U^{d_{0,1},d_{0,2}}]$. □

The above Theorem is highly constructive as it inspires recursive identification (and estimation) of the causal blip functions $B_{0,1}$ and $B_{0,2}$. This will be the topic of the following section.

3.2 Identification

In this section we show that the causal blip functions are identified from the observed distribution under consistency, sequential randomization, and positivity. We start by formally stating each of these assumptions.

Definition 1. : **Consistency**

$$H_2^{A_1} = H_2$$
$$U^A = U$$

Definition 2. : **Sequential Randomization**
For every action $a \in \mathcal{A}$ assume that

$$U^a \perp A_2^{a_1} \big| H_2^{a_1}$$
$$U^a \perp A_1 \big| H_1$$

Definition 3. : **Positivity**
For every action $a \in \mathcal{A}$ and for some $\gamma > 0$ it almost surely holds that

$$\mathbb{P}\big(A_2 = a_2 \mid H_2, A_1 = a_1\big) > \gamma$$
$$\mathbb{P}\big(A_1 = a_1 \mid H_1\big) > \gamma$$

Consistency is the assumption that allows us to observe the potential outcomes that occur by chance in the historical data. The assumption subtly states that an intervention itself does not alter the outcome if the actions remain the same. Consistency is also linked to the stable-unit-treatment-value assumption [44], which states that an intervention on one observation or unit does not affect the outcome of other units, meaning that independence is preserved between the potential outcomes in an iid population.

Sequential randomization, or exchangeability, is a generalization of the assumption of no unmeasured confounders. The single-world intervention graph presented in Figure 3.1 provides an example of a structural equation model where sequential randomization holds true [41]. The assumption would no longer hold if there were an arrow from L_1 or L_2 to A_1 or A_2. Therefore, in practical terms, if we can account for all the information used to make the historical actions, we can be confident that sequential randomization holds.

Positivity is the final assumption, stating that we should observe all possible actions in all strata of the historical data. Within the causal inference and reinforcement learning literature, this is also known as overlap or coverage. Dealing with positivity violations or near positivity violations is a major topic discussed in Manuscript Paper B. We will revisit this topic later in the chapter.

Under the above-stated assumptions, it is possible to identify the causal blip functions in two ways, both of which are constructive for the later estimation procedure. Therefore, for completeness, we will state both methods. The first result relies on the fact that the distribution of the potential observation O^a is absolutely continuous with respect to the observed distribution. For convenience, we define the g-functions as follows:

$$g_{0,k}(h_k, a_k) = \mathbb{P}(A_k = a_k | H_k = h_k) \quad k \in \{1, 2\}.$$

Then, for any $a \in \mathcal{A}$ it holds that

$$\begin{aligned}
&\mathbb{E}\left[\frac{I\{A_2 = a_2\}}{g_{0,2}(H_2, A_2)} U \Big| A_1 = a_1, V_2\right] \\
=&\mathbb{E}\left[\frac{I\{A_2^{a_1} = a_2\}}{g_{0,2}(H_2^{a_1}, a_2)} U^{a_1,a_2} \Big| V_2^{a_1}\right] \\
=&\mathbb{E}\left[\frac{\mathbb{E}[I\{A_2 = a_2\}|H_2^{a_1}]}{g_{0,2}(H_2^{a_1}, a_2)} U^{a_1,a_2} \Big| V_2^{a_1}\right] \\
=&\mathbb{E}\left[\frac{\mathbb{E}[I\{A_2 = a_2\}|H_2]}{g_{0,2}(H_2, a_2)} U^{a_1,a_2} \Big| A_1 = a_1, V_2\right] \\
=&\mathbb{E}\left[U^{a_1,a_2} \Big| V_2^{a_1}\right].
\end{aligned}$$

Note that the first expression is well defined due to positivity. The first, third, and fourth equalities hold due to consistency, and the second equality holds due to sequential randomization. Thus, we can identify the second-stage causal blip function $B_{0,2}$ and the associated second-stage optimal policy $d_{0,2}$. Now, for any a_1, we see that

$$\begin{aligned}
&\mathbb{E}\left[\frac{I\{A_1 = a_1\}}{g_{0,1}(H_1, A_1)} \frac{I\{A_2 = d_{0,2}(A_1, V_2)\}}{g_{0,2}(H_2, A_2)} U \Big| V_1\right] \\
=&\mathbb{E}\left[\frac{I\{A_1 = a_1\}}{g_{0,1}(H_1, a_1)} \frac{I\{A_2 = d_{0,2}(a_1, V_2^{a_1})\}}{g_{0,2}(H_2^{a_1}, d_{0,2}(a_1, V_2^{a_1}))} U^{a_1,a_2} \Big| V_1\right] \\
=&\mathbb{E}\left[\frac{I\{A_1 = a_1\}}{g_{0,1}(H_1, a_1)} \frac{\mathbb{E}[I\{A_2 = d_{0,2}(a_1, V_2^{a_1})\}|H_2^{a_1}]}{g_{0,2}(H_2^{a_1}, d_{0,2}(a_1, V_2^{a_1}))} U^{a_1,d_{0,2}} \Big| V_1\right] \\
=&\mathbb{E}\left[\frac{I\{A_1 = a_1\}}{g_{0,1}(H_1, a_1)} U^{a_1,d_{0,2}} \Big| V_1\right] \\
=&\mathbb{E}\left[U^{a_1,d_{0,2}} \Big| V_1\right].
\end{aligned}$$

Thus, we have identified the first-stage causal blip function along with the first-stage optimal policy. The second approach to identifying the causal blip functions is an adaptation of Q-learning. For this purpose, define the Q-function at stage 2 as follows:

$$Q_{0,2}(h_2, a_2) = \mathbb{E}[U|H_2 = h_2, A_2 = a_2]$$

Again, using consistency and sequential randomization it is possible to show that for any $a \in \mathcal{A}$

$$\mathbb{E}\big[Q_{0,2}(H_2, a_2)|A_1 = a_1, V_2\big] = \mathbb{E}[U^{a_1,a_2}|V_2^{a_1}].$$

Given the second stage optimal policy we can define the first stage Q function as follows:

$$\begin{aligned}
Q_{0,1}(h_1, a_1) &= \mathbb{E}[Q_{0,2}(H_2, d_{0,2}(V_2))|H_1 = h_1, A_1 = a_1] \\
&= \mathbb{E}[\mathbb{E}[U^{a_1,d_{0,2}}|H_2^{a_1}]|H_1 = h_1, A_1 = a_1] \\
&= \mathbb{E}[U^{a_1,d_{0,2}}|H_1 = h_1].
\end{aligned}$$

Evidently,

$$\mathbb{E}\left[Q_{0,1}(H_1, a_1)|V_1\right] = \mathbb{E}[U^{a_1,d_{0,2}}|V_1].$$

3.3 Estimation

In the previous section, we demonstrated recursive identification of the V-restricted optimal policy using the g-functions and Q-functions. This section will combine these results to construct a doubly robust loss function for the optimal policy. This loss function and other comparable loss functions will serve as the foundation for recursively learning the optimal policy.

At stage two, drawing inspiration from the efficient influence function for the single-stage static policy value in Example 2, we define the doubly robust blip function score as follows:

$$\begin{aligned} D_2(g, Q)(O) =& \frac{2A_2 - 1}{g_2(H_2, A_2)} \{U - Q_2(H_2, A_2)\} \\ &+ Q_2(H_2, 1) - Q_2(H_2, 0). \end{aligned}$$

The score is doubly robust in the sense that under the assumptions of consistency, sequential randomization, and positivity, if either $g = g_0$ or $Q = Q_0$, then

$$\mathbb{E}[D_2(g, Q)(O)|A_1, V_2] = B_{0,2}(A_1, V_2).$$

Now, for a measurable function B_2, define the doubly robust blip loss function as

$$L_2(B_2)(g, Q)(O) = \left\{D_2(g, Q)(O) - B_2(A_1, V_2)\right\}^2.$$

If either $g = g_0$ or $Q = Q_0$, the expectation of the loss function is given by

$$\begin{aligned} \mathbb{E}[L_2(B_2)(g, Q)(O)] =& \mathbb{E}\left[\left\{D_2(g, Q)(O) - B_2(A_1, V_2)\right\}^2\right] \\ =& \mathbb{E}\left[D_2(g, Q)(O)^2\right] \\ &+ \mathbb{E}\left[B_2(A_1, V_2)^2\right] \\ &- 2\mathbb{E}\left[D_2(g, Q)(O) B_2(A_1, V_2)\right] \\ =& \mathbb{E}\left[D_2(g, Q)(O)^2\right] \\ &+ \mathbb{E}\left[B_2(A_1, V_2)^2\right] \\ &- 2\mathbb{E}\left[B_{0,2}(A_1, V_2) B_2(A_1, V_2)\right] \\ =& \mathbb{E}\left[\left\{B_2(A_1, V_2) - B_{0,2}(A_1, V_2)\right\}^2\right] \\ &+ \mathbb{E}\left[D_2(g, Q)(O)^2\right] \\ &- \mathbb{E}\left[B_{0,2}(A_1, V_2)^2\right]. \end{aligned}$$

The last two terms are constant in B_2. Thus, the true second-stage blip function $B_{0,2}$ minimizes the expected blip loss. Given second-stage nuisance function estimates $\hat{g}_n$ and $\hat{Q}_n$, we can use any least square regression type estimator for the second-stage blip function denoted $\hat{B}_{2,n}$, which minimizes the empirical doubly robust blip loss:

$$\sum_{i=1}^{n} L_2(B_2)(\hat{g}_n, \hat{Q}_n)(O_i).$$

For a given second-stage policy d_2, it is again possible to construct a doubly robust loss function for the first-stage blip. Define

$$\begin{aligned} D_1(g, Q^{d_2}, d_2)(O) = & \frac{I\{A_2 = d_2(A_1, V_2)\}}{g_2(H_2, A_2)} \frac{2A_1 - 1}{g_1(H_1, A_1)} \{U - Q_2(H_2, A_2)\} \\ & + \frac{2A_1 - 1}{g_1(H_1, A_1)} \Big\{ Q_2(H_2, d_2(V_2, A_1)) - Q_1^{d_2}(H_1, A_1) \Big\} \\ & + Q_1^{d_2}(H_1, 1) - Q_1^{d_2}(H_1, 0). \end{aligned}$$

We note that

$$\begin{aligned} Q_{0,1}^{d_2}(H_1, a_1) = & \mathbb{E}\big[Q_{0,2}(H_2, d_2(A_1, V_2))\big|H_1, A_1 = a_1\big] \\ = & \mathbb{E}\big[\mathbb{E}\big[U\big|H_2, A_2 = d_2(V_2, A_1)\big]\big|H_1, A_1 = a_1\big] \\ = & \mathbb{E}\big[\mathbb{E}\big[U^{d_2}\big|H_2\big]\big|H_1, A_1 = a_1\big] \\ = & \mathbb{E}\big[U^{A_1 = a_1, d_2}\big|H_1\big]. \end{aligned}$$

Thus, if the optimal second stage policy $d_{0,2}$ is known and $Q^{d_{0,2}} = Q_0^{d_{0,2}}$, we see that

$$\mathbb{E}\left[D_1(g, Q^{d_{0,2}}, d_{0,2})(O)\big|V_1\right] = B_{0,1}(V_1).$$

Similarly, if $g = g_0$, then

$$\begin{aligned}
\mathbb{E}\left[D_1(g, Q^{d_{0,2}}, d_{0,2})(O)\middle|V_1\right] &= \mathbb{E}\left[Q_1^{d_{0,2}}(H_1,1) - Q_1^{d_{0,2}}(H_1,0)\middle|V_1\right] \\
&\quad + B_{0,1}(V_1) \\
&\quad - \mathbb{E}\left[\frac{I\{A_2 = d_2(A_1,V_2)\}}{g_2(H_2,A_2)}\frac{2A_1-1}{g_1(H_1,A_1)}Q_2(H_2,A_2)\middle|V_1\right] \\
&\quad + \mathbb{E}\left[\frac{2A_1-1}{g_1(H_1,A_1)}Q_2(H_2,d_{0,2}(V_2,A_1))\middle|V_1\right] \\
&\quad - \mathbb{E}\left[\frac{2A_1-1}{g_1(H_1,A_1)}Q_1^{d_{0,2}}(H_1,A_1)\middle|V_1\right] \\
&= E\left[Q_1^{d_{0,2}}(H_1,1) - Q_1^{d_{0,2}}(H_1,0)\middle|V_1\right] \\
&\quad + B_{0,1}(V_1) \\
&\quad - \mathbb{E}\left[\frac{2A_1-1}{g_1(H_1,A_1)}Q_2(H_2,d_{0,2}(V_2,A_1))\middle|V_1\right] \\
&\quad + \mathbb{E}\left[\frac{2A_1-1}{g_1(H_1,A_1)}Q_2(H_2,d_{0,2}(V_2,A_1))\middle|V_1\right] \\
&\quad - \mathbb{E}\left[Q_1^{d_{0,2}}(H_1,1) - Q_1^{d_{0,2}}(H_1,0)\middle|V_1\right] \\
&= B_{0,1}(V_1).
\end{aligned}$$

Combining the above results yields that if either $g = g_0$ or $Q^{d_{0,2}} = Q_0^{d_{0,2}}$, a valid loss function is given by

$$L_1(B_1)(g, Q^{d_{0,2}}, d_{0,2})(O) = \left\{D_1(g, Q^{d_{0,2}}, d_{0,2})(O) - B_1(V_1)\right\}^2.$$

In practice, for an estimated second-stage optimal policy $\hat{d}_{n,2}$ and nuisance function estimates $\hat{g}_n$ and $\hat{Q}_n^{\hat{d}_{n,2}}$, we can use any least square regression type estimator for the first-stage blip function, denoted $\hat{B}_{1,n}$, which minimizes the empirical doubly robust blip loss:

$$\sum_{i=1}^{n} L_1(B_2)(\hat{g}_n, \hat{Q}_n^{\hat{d}_{n,2}}, \hat{d}_{n,2})(O_i).$$

In the single-stage case, cross-fitting the nuisance functions can be done easily following the approach described for the one-step estimator in Chapter 2. However, when dealing with two stages or more, it becomes more challenging to cross-fit the first-stage Q-function. This is because the estimation of the first-stage Q-function depends on the estimated second-stage optimal policy. Ideally, a nested cross-fitting scheme should be applied to handle this situation correctly.

The least square blip loss function is just one way to frame the optimal policy problem. Other notable alternative loss functions include the value loss

function, which leads to value search or partitioning procedures [57, 62, 23], and the weighted classification loss function, which leads to (augmented) outcome weighted learning [60, 32, 20, 58]. These value loss functions and classification loss functions are presented in detail in Paper A. Both the value loss and the classification loss can be seen as more direct approaches focusing on the decision boundary. In contrast, learning the blips focuses on the estimation error of the blip functions rather than just the associated decision boundary.

As mentioned by [30], an advantage of estimating the blip functions is that it allows us to identify subgroups of patients for which following the estimated optimal policy is particularly beneficial. This is easy to see in the single stage case. Given a threshold $\eta > 0$, we can define the subgroup indicator as follows:

$$s_\eta(V_1) = I\left\{B_{0,1}(V_1) > \eta\right\}.$$

Then, the above subgroup indicator identifies the group of patients with a conditional average treatment effect of at least η. The associated subgroup average treatment effect is then given by:

$$\mathbb{E}[U^1 - U^0 | s_\eta(V_1) = 1].$$

The concept of subgroups can be generalized to the multi-stage case, where we compare the expected utility outcome under the optimal policy with the outcome under a reference policy.

The `polle` R package, as described in Paper A, unifies many of the available doubly robust policy learners on CRAN [1] as well as introducing doubly robust blip learning. The function `policy_learn()` is used to specify a given policy learner. Table 3.1 provides an overview of the available policy learning methods. Unless specified otherwise, the nuisance function values used to construct the doubly robust score at each stage can be cross-fitted.

3.4 Performance

In this section, we introduce various performance measures for policy learning. For simplicity, we consider a single-stage setup with

$$O = (H, A, U),$$

where A is binary. Under consistency, sequential randomization, and positivity, an unrestricted optimal policy is given by

$$d_0(h) = I\Big\{B_0(h) > 0\Big\}$$
$$B_0(h) = Q_0(h,1) - Q_0(h,0),$$

[1] https://cran.r-project.org/web/views/CausalInference.html

`type` argument	Method	Imports	Limitations
`"ql"`	Q-learning		
`"drql"`	Doubly Robust Q-learning		
`"blip"`	Doubly Robust blip-learning		Only available for dichotomous action sets.
`"ptl"`	Policy tree learning	`policytree`	Realistic policy learning implemented for dichotomous action sets.
`"owl"`	Outcome weighted learning	`DTRlearn2`	No realistic policy learning. Fixed number of stages. Dichotomous action set. Augmentation terms are not cross-fitted.
`"earl"`	Efficient augmented and relaxation learning	`DynTxRegime`	Single stage. No cross-fitting. No realistic policy learning. Dichotomous action set.
`"rwl"`	Residual weighted learning	`DynTxRegime`	Same as `"earl"`.

Table 3.1: Overview of policy learning methods and their dependencies and limitations.

where B_0 is the blip function. An obvious target parameter that can be used to measure the maximum achievable performance is the optimal policy value given by

$$\Psi(P_0) = \mathbb{E}\left[Q_0(H, d_0(H))\right].$$

Inference for the optimal policy value has generally been challenging due to the parameter's inherent non-smooth structure around the decision boundary where the blip function is zero. This challenge was initially highlighted by [42]. To ensure the existence of an influence function, we use the non-exceptional law, which states that the blip function is almost surely bounded away from zero. Note that this assumption implies that the optimal policy is unique.

Definition 4. : **Non-exceptional law**
For some $\delta > 0$ it holds that

$$\mathbb{P}\big(|B_0(H)| > \delta\big) = 1.$$

The non-exceptional law is a strong assumption that can be challenging to justify, particularly in settings with treatment non-responders.

The following technical Lemma is helpful when dealing with model-based threshold policies:

Lemma 4. *for some x and y on $\mathbb{R}$*

$$|I\{x > 0\} - I\{y > 0\}| \leq I\{|y| \leq |x - y|\} \tag{3.2}$$

Proof. If $x > 0$ and $y > 0$ or $x \leq 0$ and $y \leq 0$, then the statement is trivial. If $x > 0$ and $y \leq 0$ then $-(x - y) < y \leq 0$. Similarly, if $y > 0$ and $x \leq 0$ then $(x - y) \leq -y < 0$. □

The following Theorem can be found in [34], see also [19]. We also give a proof for completeness and because it is instructive for working with pathwise derivatives.

Theorem 5. *Assume that, for some constant $C < \infty$, $\mathbb{P}(|U| < C) = 1$ and $|Q_0(H, a)| < C$ almost surely for $a \in \{0, 1\}$. Furthermore, assume that the non-exceptional law holds. Then $\Psi(P_0)$ is pathwise differentiable with efficient influence function*

$$\psi(g_0, Q_0, d_0)(O) = \frac{I\{A = d_0(H)\}}{g_0(H, A)} \{U - Q_0(H, d_0(H)\} + Q_0(H, d_0(H)) - \Psi(P_0). \tag{3.3}$$

Proof. As we will see, the target parameter is not sensitive to fluctuations of the action model $f_0(a \mid h) = g_0(h, a)$. Thus, the relevant tangent space consists of the closure of the direct sum of the sub-tangent-spaces

$$\begin{aligned}
\overline{T}_H(P_0) &= \{s(H) \in L_2(P_0) : \mathbb{E}[s(H)] = 0\} \\
\overline{T}_U(P_0) &= \{s(H, A, U) \in L_2(P_0) : \mathbb{E}[s(H, A, U)|H, A] = 0\}.
\end{aligned}$$

To construct a valid parametric submodel, we only consider scores bounded by C. A parametric submodel is then given by

$$\begin{aligned}
f_\epsilon(h) &= (1 + \epsilon s(h)) f_0(h) \\
f_\epsilon(u \mid h, a) &= (1 + \epsilon s(h, a, u)) f_0(u \mid h, a).
\end{aligned}$$

The target parameter can be rewritten as

$$\begin{aligned}
\Psi(P_0) = &\mathbb{E}\left[d_0(h) B_0(H)\right] \\
&+ \mathbb{E}\left[Q_0(H, 0)\right].
\end{aligned}$$

We directly calculate the pathwise derivative, where we note that the optimal policy under the parametric submodel is given by

$$d_\epsilon(h) = I\{B_\epsilon(h) > 0\}. \tag{3.4}$$

Thus,

$$
\begin{aligned}
&\epsilon^{-1}\left\{\Psi(P_\epsilon) - \Psi(P_0)\right\} \\
&= \epsilon^{-1}\int d_\epsilon(h)\, B_\epsilon(h) f_{0,\epsilon}(h) d\nu(u) + \epsilon^{-1}\int u\, f_{0,\epsilon}(u \mid h, 0) f_{0,\epsilon}(h) d\nu(u,h) \\
&\quad - \epsilon^{-1}\int d_0(h)\, B_0(h) f_0(h) d\nu(h) - \epsilon^{-1}\int u\, f_0(u \mid h, 0) f_0(h) d\nu(u,h) \\
&= \epsilon^{-1}\int \left\{d_\epsilon(h) - d_0(h)\right\}\, B_\epsilon(h) f_\epsilon(h) d\nu(h) && (3.5) \\
&\quad + \epsilon^{-1}\int d_0(h)\, \left\{B_\epsilon(h) f_\epsilon(h) - B_0(h) f_0(h)\right\} d\nu(h) && (3.6) \\
&\quad + \epsilon^{-1}\int u\, \left\{f_\epsilon(u \mid h, 0) f_\epsilon(h) - f_0(u \mid h, 0) f_0(h)\right\} d\nu(u,h). && (3.7)
\end{aligned}
$$

The expression in (3.6) and (3.7) corresponds to the pathwise derivative of $\mathbb{E}[Q_0(H, d(H))]$ for a known policy $d = d_0$ with associated efficient influence function $\psi(g_0, Q_0, d_0)(O)$, see Example 2. It is left to prove that the remaining term (3.5) converges to zero as ϵ approaches zero. For this purpose, we rewrite the fluctuated blip as

$$
\begin{aligned}
&B_\epsilon(h) \\
&= \int u \left\{f_\epsilon(u \mid h, 1) - f_\epsilon(u \mid h, 0)\right\} d\nu(u) \\
&= \int u \left\{(1 + \epsilon s(h,1,u)) f_0(u \mid h, 1) - (1 + \epsilon s(h,0,u)) f_0(u \mid h, 0)\right\} d\nu(u) \\
&= B_0(h) + \epsilon \int u\, \left\{s(h,1,u) f_0(u \mid h, 1) - s(h,0,u) f_0(u \mid h, 0)\right\} d\nu(u), && (3.8)
\end{aligned}
$$

where, by assumption, the last term is bounded because

$$
\left| \int u\, \left\{s(h,1,u) f_0(u \mid h, 1) - s(h,0,u) f_0(u \mid h, 0)\right\} d\nu(u) \right| \leq 2C^2, \qquad (3.9)
$$

Going back to expression (3.5), we see that

$$
\begin{aligned}
&\epsilon^{-1}\left| \int \left\{d_\epsilon(h) - d_0(h)\right\}\, B_\epsilon(h) f_\epsilon(h) d\nu(h) \right| \\
&\leq \int |d_\epsilon(h) - d_0(h)|\, |B_\epsilon(h)|\, f_\epsilon(h) d\nu(h) \\
&\leq \epsilon^{-1}\int I\{|B_0(h)| \leq |B_\epsilon(H) - B_0(h)|\}\, |B_\epsilon(h)|\, f_\epsilon(h) d\nu(h) && (3.10) \\
&\leq \epsilon^{-1}\int I\{|B_0(h)| \leq \epsilon 2C^2\}\, \left\{|B_0(h)| + \epsilon 2C^2\right\} \left\{1 + \epsilon C\right\} f_0(h) d\nu(h) && (3.11) \\
&\leq \epsilon^{-1}\int I\{|B_0(h)| \leq \epsilon 2C^2\}\, \left\{\epsilon 2C^2 + \epsilon 2C^2\right\} \left\{1 + \epsilon C\right\} f_0(h) d\nu(h) \\
&\leq\ 4C^2 \left\{1 + \epsilon C\right\} \int I\{\delta < |B_0(h)| \leq \epsilon 2C^2\} f_0(h) d\nu(h). && (3.12)
\end{aligned}
$$

Inequality (3.10) follows by Lemma 4, (3.11) follows by (3.8), (3.9), and the definition of the bounded parametric submodel, and (3.12) follows by the non-exceptional law. Finally, by dominated convergence, the last expression (3.12) goes to zero as ϵ goes to zero. □

As also used in the proof of Theorem 5, we recognize expression (3.3) as the efficient influence function for the value of the optimal policy, assuming that we already know the optimal policy d_0. At first glance, it appears that we do not pay the price asymptotically by first having to estimate the optimal policy. However, our hope for good fortune is soon to end. We must estimate the Q-function (and the policy) at a parametric model rate $n^{1/2}$ to construct an estimator with the above influence function. This result follows from studying the second-order remainder:

Theorem 6. *Given a policy d and nuisance functions Q and g, where $g(H, a) > \gamma$ almost surely for some $\gamma > 0$, the second order remainder for the efficient influence function of the optimal policy value is given by*

$$\begin{aligned}
&R(g, Q, d, g_0, Q_0, d_0) \\
&= \mathbb{E}\left[\frac{g_0(H, d(H)) - g(H, d(H))}{g(H, d(H))} \{Q_0(H, d(H)) - Q(H, d(H))\}\right] \\
&\quad + \mathbb{E}\left[\{d(H) - d_0(H)\} B_0(H)\right] \\
&= R_1(g, Q, d, g_0, Q_0) \\
&\quad + R_2(Q, d, Q_0, d_0).
\end{aligned}$$

Proof. By definition of the second order remainder term:

$$\begin{aligned}
R(g,Q,d,g_0,Q_0,d_0) &= \mathbb{E}[\psi(g,Q,d)(O)] + \Psi(Q,g,d) - \Psi(P_0) \\
&= \mathbb{E}\left[\frac{I\{A=d(H)\}}{g(H,A)}\{U - Q(H,A)\}\right] \\
&\quad + \mathbb{E}[Q(H,d(H))] - \Psi(P_0) \\
&= \mathbb{E}\left[\frac{I\{A=d(H)\}}{g(H,d(H))}\{Q_0(H,d(H)) - Q(H,d(H))\}\right] \\
&\quad + \mathbb{E}[Q(H,d(H))] - \Psi(P_0) \\
&= \mathbb{E}\left[\frac{g_0(H,d(H))}{g(H,d(H))}\{Q_0(H,d(H)) - Q(H,d(H))\}\right] \\
&\quad + \mathbb{E}[Q(H,d(H))] - \Psi(P_0) \\
&= \mathbb{E}\left[\frac{g_0(H,d(H))}{g(H,d(H))}\{Q_0(H,d(H)) - Q(H,d(H))\}\right] \\
&\quad + \mathbb{E}[Q(H,d(H))] \\
&\quad + \mathbb{E}[Q_0(H,d(H))] - \mathbb{E}[Q_0(H,d(H))] - \Psi(P_0) \\
&= \mathbb{E}\left[\frac{g_0(H,d(H)) - g(H,d(H))}{g(H,d(H))}\{Q_0(H,d(H)) - Q(H,d(H))\}\right] \\
&\quad + \mathbb{E}[Q_0(H,d(H))] - \Psi(P_0) \\
&= \mathbb{E}\left[\frac{g_0(H,d(H)) - g(H,d(H))}{g(H,d(H))}\{Q_0(H,d(H)) - Q(H,d(H))\}\right] \\
&\quad + \mathbb{E}[Q_0(H,0) + d(H)B_0(H)] - \Psi(P_0) \\
&= \mathbb{E}\left[\frac{g_0(H,d(H)) - g(H,d(H))}{g(H,d(H))}\{Q_0(H,d(H)) - Q(H,d(H))\}\right] \\
&\quad + \mathbb{E}\left[\{d(H) - d_0(H)\}B_0(H)\right],
\end{aligned}$$

□

The first remainder term R_1 is the typical product remainder term, which, under positivity, can be bounded up to a constant by the product of the nuisance function $L_2(P_0)$ errors:

$$\|Q_0(H,A) - Q(H,A)\|_{2,P_0} \times \|g_0(H,A) - g(H,A)\|_{2,P_0}$$

Hence, a convergence rate of $n^{1/4}$ for both the Q-function and g-function estimates will be sufficient to ensure a convergence rate of $n^{1/2}$ for the remainder term R_1. The second remainder term R_2 poses more of a challenge. Letting d be the threshold policy associated with Q given by

$$d(H) = I\{B(H) > 0\} = I\{Q(H,1) - Q(H,0) > 0\},$$

we can again use Lemma 4 to bound R_2:

$$\begin{aligned}
&\|\{d(H) - d_0(H)\} B_0(H)\|_{2,P_0} \\
\leq &\|I\{|B_0(H)| \leq |B(H) - B_0(H)|\}|B_0(H)|\|_{2,P_0} \\
\leq &\|B(H) - B_0(H)\|_{2,P_0}
\end{aligned}$$

Thus, to achieve a convergence rate of $n^{1/2}$ for the second-order remainder R, it is necessary to estimate the blip function and, in turn, the Q-function at a parametric rate. This requirement, combined with the necessity of the non-exceptional law, means that asymptotic bounds for the optimal policy value are not suitable for measuring the maximum performance of a policy learner.

Instead, a part of the literature has focused on bounding the expected regret [4, 18, 40, 3]. For a given policy within a class of policies $\mathcal{D}$, the regret is defined as the difference between the value of the optimal policy within the class and the value of the chosen policy:

$$\text{regret}(d) = \max_{d' \in \mathcal{D}} \left\{ \mathbb{E}\left[Q_0(H, d'(H))\right] \right\} - \mathbb{E}\left[Q_0(H, d(H))\right]$$

For a policy estimate $\hat{d}_n$, the expected regret over samples of size n is given by $\mathbb{E}[\text{regret}(\hat{d}_n)]$. Although putting asymptotic bounds on the expected regret of a policy learner is more of a theoretical performance guarantee than a practical measure for evaluating a policy learner, it is still important as it allows us to construct well-behaved policy estimators.

As shown in [3], bounding the regret of an unrestricted class of policies is too ambitious. Following the key insights of [29], the authors focus on classes of policies with a bounded Vapnik-Chervonenkis (VC) dimension. The VC dimension of the policy class can grow with the sample size, but not too quickly. Examples of policy classes with bounded VC dimensions include linear threshold policies and decision trees.

Even under policy class restrictions, not all policy estimators are equally suited. We need policy value estimators that are strong enough to withstand optimization over the class of policies. In this regard, the authors of [3] study a policy learner that empirically minimizes the doubly robust value loss. For this policy estimator, they achieve optimal $n^{1/2}$ bounds for the regret. The proof critically depends on the efficiency of the doubly robust value estimator.

In our view, restricting the policy class is preferable to making further distributional assumptions. A notable example of such assumptions that can lead to a bounded regret is presented in [4]. This approach, similar to the non-exceptional law, limits the concentration of the blip function around zero through margin conditions. As mentioned, while performance guarantees based on regret bounds are relevant for theoretical comparisons of policy learners, they may not be helpful in practical situations for policymakers to analyze the gain in value and associated risk of implementing an estimated

policy. Instead, we strongly advocate for targeting the estimated policy's value [12]. Conditional on $\hat{d}_n$, the target parameter is given by:

$$\Psi^{\hat{d}_n}(P_0) = \mathbb{E}[U^{\hat{d}_n}].$$

This data-adaptive policy value is arguably more practically relevant than the true optimal policy value because the true optimal policy will never actually be implemented. Moreover, as we will see, the data-adaptive policy value does not suffer from non-regularity issues, eliminating the need for the non-exceptional law. Let the data-adaptive one-step estimator be given by

$$\hat{\Psi}_n^{\hat{d}_n} = n^{-1} \sum_{i=1}^{n} \left(\frac{I\{A_i = \hat{d}_n(H_i)\}}{\hat{g}_n(H_i, A_i)} \left\{ U_i - \hat{Q}_n(H_i, \hat{d}_n(H_i)) \right\} + \hat{Q}_n(H_i, \hat{d}_n(i)) \right).$$

If, for some (limiting) policy d' [52], it holds that

$$\{P_n - P_0\} \left\{ \psi(\hat{g}_n, \hat{Q}_n, \hat{d}_n)(X) - \psi(g_0, Q_0, d')(X) \right\} = o_{P_0}(n^{-1/2}), \tag{3.13}$$

a direct application of the von Mises expansion yields that

$$\begin{aligned} \hat{\Psi}_n^{\hat{d}_n} - \Psi^{\hat{d}_n}(P_0) = n^{-1} \sum_{i=1}^{n} \psi(g_0, Q_0, d')(X_i) \\ + R_1(\hat{g}_n, \hat{Q}_n, \hat{d}_n, g_0, Q_0) \\ + o_{P_0}(n^{-1/2}). \end{aligned}$$

As described, under positivity, the second order remainder term R_1 is $o_{P_0}(n^{-1/2})$ if conditional on $\hat{Q}_n$ and $\hat{g}_n$ it holds that

$$\|Q_0(H, A) - \hat{Q}_n(H, A)\|_{2,P_0} \times \|g_0(H, A) - \hat{g}_n(H, A)\|_{2,P_0} = o_{P_0}(n^{-1/2}). \tag{3.14}$$

We are left to show that the empirical process remainder term is negligible under reasonable conditions, i.e., that (3.13) holds. In Appendix I, we show that if $\hat{Q}_n$, $\hat{g}_n$, and $\hat{d}_n$ falls in a Donsker class with probability approaching one, and conditional on $\hat{Q}_n$, $\hat{g}_n$, and $\hat{d}_n$ it holds that

$$\|\hat{Q}_n(H, A) - Q_0(H, A)\|_{2,P_0} = o_{P_0}(1), \tag{3.15}$$

$$\|\hat{g}_n(H, A) - g_0(H, A)\|_{2,P_0} = o_{P_0}(1), \tag{3.16}$$

$$P_0|\hat{d}_n(H) - d'(H)| = o_{P_0}(1), \tag{3.17}$$

then (3.13) holds true. Ideally, $d' = d_0$, but this does not have to be the case. Of course, under an exceptional law, we may still be critical whether $\hat{d}_n$ converges towards a fixed policy at any rate.

We further advocate for constructing a cross-fitted version of the data-adaptive policy value estimate [45], where the g-function, Q-function, and

policy are estimated on each training fold, and the doubly robust score is calculated on each validation fold. The details of this procedure are described in Paper A and implemented in the `polle` R package via the `policy_eval()` function. To our knowledge, this very important performance measure is not available in any other software implementation.

To conclude this section, we would like to add some notes on the multi-stage case. As of our current knowledge, regret bounds like those found in [3] have yet to be generalized to the multi-stage case. However, the data-adaptive policy value estimator remains asymptotically linear under conditions similar to those presented in this section. We will not delve into the details, but refer to [34] for further information.

3.5 Positivity violations

The assumptions of consistency and sequential randomization are untestable, meaning we cannot directly verify them based on the observed data. However, the assumption of positivity is a property of the observed distribution, which enables us to identify violations or practical violations [6, 38].

Despite the importance of the positivity assumption, issues related to positivity violations have received surprisingly limited attention in the statistical policy learning literature. A retargeting approach via a weighted value function has been suggested by [24]. Among the reviews [48, 30, 13], only [13] mentions that methods for handling positivity violations in multi-stage settings are underdeveloped. Positivity violations are also known as partial coverage in the offline reinforcement literature. For recent reviews on off-policy reinforcement learning and challenges related to partial coverage in Markov decision problems, we refer to [49, 50].

Practical positivity violations are a considerable issue in the application of Paper B, as also described in Chapter 4. This issue further motivated the development of the `polle` R package Paper A, as no other software implementations of statistical policy learning offer methods for dealing with practical positivity violations. We advocate for changing the targeted policy. Instead of targeting the globally optimal policy, we target the optimal policy within the set of policies deemed realistic at a given level $\alpha > 0$. In the single-stage binary action case, the globally optimal policy is given by the zero threshold policy associated with the blip function. Let d_0^α denote the realistic modification of d_0:

$$d_0^\alpha(h) = I\{g_0(h,1) \in (\alpha, 1-\alpha)\} I\{B_0(h) > 0\} + I\{g_0(h,1) \in [1-\alpha, 1)\}.$$

By construction, d_0^α will not lead to positivity violations in the sense that $g_0(H, d_0^\alpha(H)) \geq \alpha$ almost surely. Let $\hat{d}_n^\alpha$ denote the associated plug-in estimator, which, if the true g-function is unknown, will depend on the estimate

$\hat{g}_n$. Realistic policy learning is available in the polle package via the `alpha` argument in the `policy_learn()` function.

Performance guarantees in terms of regret for this policy estimator have, to our knowledge, not been studied in the literature. However, the data-adaptive policy value estimator can easily accommodate the above modification. Let

$$G_0(h) = \{a \in \mathcal{A} : g_0(h, a) > 0\}$$

denote the feasible set of actions for a given h. We assume positivity for the feasible set:

$$g_0(H, a) > \gamma \quad \forall\ a \in G_0(H) \quad a.s.$$

We want to verify that under positivity for the feasible set, the convergence conditions (3.14), (3.15), (3.16), and (3.17) still lead to asymptotic linearity of the realistic data-adaptive policy value estimator. Without loss of generality, we assume that $\alpha > \gamma$. Because (3.16) holds, it also holds with probability tending one that

$$\hat{d}_n^\alpha(H) \in G_0(H).$$

Thus, conditional on $\hat{Q}_n$ and $\hat{d}_n^\alpha$ we see that

$$\begin{aligned}
&\|\hat{Q}_n(H, A) - Q_0(H, A)\|_{2,P_0}^2 \\
&= \mathbb{E}\left[\left\{\hat{Q}_n(H,1) - Q_0(H,1)\right\}^2 g_0(H,1) + \left\{\hat{Q}_n(H,0) - Q_0(H,0)\right\}^2 g_0(H,0)\right] \\
&\geq \gamma\ \mathbb{E}\left[\left\{\hat{Q}_n(H,1) - Q_0(H,1)\right\}^2 I\{1 \in G_0(H)\} + \left\{\hat{Q}_n(H,0) - Q_0(H,0)\right\}^2 I\{0 \in G_0(H)\}\right] \\
&\geq \gamma\ \mathbb{E}\Big[\left\{\hat{Q}_n(H,1) - Q_0(H,1)\right\}^2 I\{1 \in G_0(H)\} I\{\hat{d}_n(H) = 1\} \\
&\qquad + \left\{\hat{Q}_n(H,0) - Q_0(H,0)\right\}^2 I\{0 \in G_0(H)\} I\{\hat{d}_n(H) = 0\}\Big] \\
&= \gamma\ \|\hat{Q}_n(H, \hat{d}_n(H)) - Q_0(H, \hat{d}_n(H))\|_{2,P_0}^2.
\end{aligned}$$

Similarly, with probability approaching one

$$\begin{aligned}
&\|\hat{g}_n(H, A) - g_0(H, A)\|_{2,P_0}^2 \\
&\geq \gamma\ \|\hat{g}_n(H, \hat{d}_n(H)) - g_0(H, \hat{d}_n(H))\|_{2,P_0}^2.
\end{aligned}$$

Hence, both the second order remainder term and the empirical process remainder term converge to zero at the required rates. We can conclude that the realistic data-adaptive policy value estimator is asymptotically linear under positivity for the feasible set.

3.6 Stochastic number of stages

In many applications, including the case addressed in Paper B, the decision-making process is influenced by an underlying marked point process, where

the timing between decisions naturally varies. For example, in the refrigerated container maintenance and repair problem presented in Chapter 4, the timing between breakdowns will vary, resulting in a varying number of stages within a given time interval for each observation. Up until now, we have only considered a fixed number of stages. Fortunately, the recursive methodology developed for a fixed number of stages can be extended to handle a stochastic number of stages, assuming that the maximum number of stages is finite [16].

For simplicity, we assume that the maximum number of stages is 2. For convenience, if the state variables occur, we represent them as $S_1 = (X_1, U_1)$, $S_2 = (X_2, U_2)$, and $S_3 = U_3$. The action variables remain as A_1 and A_2, and the utility is the sum of the occurring rewards U_k. For an observation with a single decision stage, the likelihood is given by:

$$f_0(S_1)f_0(A_1|S_1)f_0(S_2|S_1, A_1),$$

where $X_2 = \emptyset$ by construction, indicating that the process has terminated. Similarly, for an observation with two decision stages, the likelihood is:

$$f_0(S_1)f_0(A_1|S_1)f_0(S_2|S_1, A_1)f_0(A_2|S_1, A_1, S_2)f_0(S_3|S_1, A_1, S_2, A_2).$$

The above decomposition of the likelihood inspires a degenerate extension of the single decision observation using the auxiliary variables.

$$A_2^* = a_2^\dagger,\ U_3^* = 0,$$

for some fixed action $a_2^\dagger \in \mathcal{A}_2$. For the resulting auxiliary observations O^* it holds that

$$g_{0,2}^*(a_2^\dagger|S_1, A_1, S_2 = \emptyset) = 1$$
$$Q_{0,2}^*(S_1, A_1, S_2 = \emptyset, A_2 = a_2^\dagger) = U_1 + U_2 + 0.$$

An adaptation of Lemma 4.1 in [16] yields that for any feasible policy d over the auxiliary distribution, it holds that $\mathbb{E}[U^d] = \mathbb{E}[U^{*,d}]$. Thus, finding the optimal policy in the auxiliary fixed-stage decision problem corresponds to finding the optimal policy in the stochastic-stage decision problem. This functionality is also implemented in the `polle` R package Paper A. To our knowledge, no other package in R has similar functionality.

3.7 Functional inference

So far, we have only been concerned with inference for the (data-adaptive) policy value. The value is arguably a key parameter, but it does not provide any information on why and how the policy works. The form and interpretability of a policy is obviously important in the medical sciences, but in an industrial

setting, it can also help build confidence in the policy and persuade any critics that the policy is sensible.

Inference for functional parameters is an exciting field of research. For example, [33] studies general differentiable Hilbert-valued parameters, including key examples such as the potential mean outcome under continuous treatment and the conditional average treatment effect. The latter example corresponds precisely to the blip in the single-stage case. Other recent works studying the estimation of heterogeneous treatment effects include [27, 28, 37]. General functional parameters tend not to be differentiable, meaning no influence functions exist. the mentioned works rely on restricting the functional target to a reproducing kernel Hilbert space, making local polynomial approximations, or regularizing the problem in other ways. These results are directly related to the lack of existence of an influence function for the value of the true optimal policy if we cannot estimate the Q-function at a sufficiently fast rate.

A general review of inference for (policy) functionals is beyond the scope of this thesis. However, we will focus on a specific line of research that involves inference for the parameters in the best smooth model approximation of the V-restricted Q-functions or blip functions. This topic is also referred to as variable importance in [51] and is advocated for by [11]. Another closely related approach, proposed by [56], considers associational parameters linked to the parameters of a generalized linear model.

Variable importance as an approximation

For simplicity, we consider a single-stage case with observations $O = (H, A, U)$, where H denotes the baseline variables, A denotes the action, and U denotes the outcome. Let V be a function of H and let $\Psi(P_0)(V)$ denote the parameter of interest, e.g., the V-restricted Q-function $\mathbb{E}[\mathbb{E}[U|A = a, H]|V]$ or just the conditional average treatment outcome. Usually, for continuous V, this parameter will not have an influence function. Thus, we consider the best approximation given by the projection

$$\beta(P_0) = \arg\min_{\beta} \mathbb{E}\left[\left(\Psi(P_0)(V) - m(V;\beta)\right)^2\right], \tag{3.18}$$

for a smooth model m with parameter β. Whereas $\Psi(P_0)(V)$ in general does not have an influence function, we aim to find an (efficient) influence function for the finite parameter $\beta_0 = \beta(P_0)$.

The Gateaux derivate gives a candidate for the efficient influence function. Under regularity conditions, $\beta(P)$ solves

$$\Omega(\beta, P) = \mathbb{E}_P\left[m_1(V;\beta)\left\{\Psi(P)(V) - m(V;\beta)\right\}\right] = 0,$$

where $m_1(V;\beta) = \frac{\partial}{\partial\beta} m(V;\beta)$. For a parametric submodel P_ϵ, the derivative is given by the implicit derivative

$$\beta(P_\epsilon)\Big|_{\epsilon=0} = -\left[\frac{\partial\Omega(\beta,P_0)}{\partial\beta}\Big|_{\beta=\beta_0}\right]^{-1} \frac{\partial\Omega(\beta_0,P_\epsilon)}{\partial\epsilon}\Big|_{\epsilon=0}. \tag{3.19}$$

We immediately have that the inverse of the first factor of (3.19) is given by

$$\frac{\partial\Omega(\beta,P_0)}{\partial\beta}\Big|_{\beta=\beta_0} = \mathbb{E}\left[m_2(V;\beta_0\{\Psi(P_0)(V) - m(V;\beta_0\} - m_1(V;\beta_0)^{\otimes 2}\right]$$

where $m_2(V;\beta) = \frac{\partial^2 m(V;\beta)}{\partial\beta^T\partial\beta}$. Define

$$h(V,P_\epsilon) = m_1(V;\beta_0)\left\{\Psi(P_\epsilon)(V) - m(V;\beta_0\right\}.$$

By Lemma 2 and the fact that $\Omega(\beta(P),P) = 0$, the second factor of (3.19) is given by

$$\begin{aligned}\frac{\partial\Omega(\beta_0,P_\epsilon)}{\partial\epsilon}\Big|_{\epsilon=0} &= \mathbb{E}\left[\frac{\partial}{\partial\epsilon}h(V,P_\epsilon)\Big|_{\epsilon=0}\right] + h(\widetilde{v},P_0) - \mathbb{E}\left\{h(V,P_0)\right\} \\ &= \mathbb{E}\left[\frac{\partial}{\partial\epsilon}h(V,P_\epsilon)\Big|_{\epsilon=0}\right] + h(\widetilde{v},P_0),\end{aligned}$$

where

$$\frac{\partial}{\partial\epsilon}h(v,P_\epsilon)\Big|_{\epsilon=0} = m_1(v;\beta_0)\,\frac{\partial}{\partial\epsilon}\Psi(P_\epsilon)(v)\Big|_{\epsilon=0}.$$

Combining the above results yields that the Gateaux derivative of β_0 is given by

$$\begin{aligned}\beta(P_\epsilon)\Big|_{\epsilon=0} &= -\mathbb{E}\Big[m_2(V;\beta_0)\{\Psi(P_0)(V) - m(V;\beta_0)\} - m_1(V;\beta_0)^{\otimes 2}\Big]^{-1} \\ \times &\left\{\mathbb{E}\left[m_1(V;\beta_0)\,\frac{\partial}{\partial\epsilon}\Psi(P_\epsilon)(V)\Big|_{\epsilon=0}\right] + m_1(\widetilde{v};\beta_0)\left\{\Psi(P_\epsilon)(\widetilde{v}) - m(\widetilde{v};\beta_0)\right\}\right\}\end{aligned} \tag{3.20}$$

Conditional average treatment outcome

As mentioned, an important example related to policy learning is the conditional average treatment outcome:

$$\Psi(P_0)(V) = \mathbb{E}[\mathbb{E}[U|A=a,H]|V],$$

which is also the V-restricted Q-function in the single-stage policy learning problem. By direct calculation, we see that the Gateaux derivative is given

by

$$\begin{aligned}
&\frac{\partial}{\partial\epsilon}\Psi(P_\epsilon)(v)\Big|_{\epsilon=0} \\
&= \int \frac{\partial}{\partial\epsilon}\mathbb{E}_{P_\epsilon}[U|A=a,H=h]\Big|_{\epsilon=0} f(h|v)dh \qquad (3.21)\\
&+ \int \mathbb{E}[U|A=a,H=h]\ \frac{\partial}{\partial\epsilon}f_\epsilon(h|v)\Big|_{\epsilon=0} dh. \qquad (3.22)
\end{aligned}$$

By Example 1, we see that line (3.21) equals

$$\begin{aligned}
&\int \frac{\partial}{\partial\epsilon}\mathbb{E}_{P_\epsilon}[U|A=a,H=h]\Big|_{\epsilon=0} f(h|v)dh \\
=&\int \frac{I(\widetilde{a}=a)I(\widetilde{h}=h)}{\mathbb{P}(A=a,H=h)}\Big\{\widetilde{u}-\mathbb{E}[U|A=a,H=h]\Big\}f(h|v)dh \\
=&\int \frac{I(\widetilde{a}=a)I(\widetilde{h}=h)}{\mathbb{P}(A=a,H=h)}\Big\{\widetilde{u}-\mathbb{E}[U|A=a,H=h]\Big\}\frac{f(h)I(V(h)=v)}{f(v)}dh \\
=&\frac{I(\widetilde{a}=a)}{\mathbb{P}(A=a|H=\widetilde{h})}\frac{I(\widetilde{v}=v)}{f(v)}\Big\{\widetilde{u}-\mathbb{E}[U|A=a,H=\widetilde{h}]\Big\}.
\end{aligned}$$

By Lemma 1, we also see that

$$\frac{\partial}{\partial\epsilon}f_\epsilon(h|v)\Big|_{\epsilon=0} = \frac{I(\widetilde{h}=h)I(\widetilde{v}=v)}{f(v)} - \frac{I(\widetilde{v}=v)f(h|v)}{f(v)}.$$

Thus line (3.22) is equal to

$$\frac{I(\widetilde{v}=v)}{f(v)}\Big\{\mathbb{E}[U|A=a,H=\widetilde{h}]-\mathbb{E}\left[\mathbb{E}[U|A=a,H]\mid V=v\right]\Big\}.$$

Combining the above results yields that

$$\begin{aligned}
&\frac{\partial}{\partial\epsilon}\Psi(P_\epsilon)(v)\Big|_{\epsilon=0} \\
&\quad= \frac{I(\widetilde{v}=v)}{f(v)}\Big\{D(P_0)(\widetilde{o})-\Psi(P_0)(\widetilde{v})\Big\},
\end{aligned}$$

where $D(P_0)$ is the doubly robust score:

$$D(P_0)(O) = \frac{I\{A=a\}}{\mathbb{P}(A=a|H)}\left[U-\mathbb{E}[U|A=a,H]\right]+\mathbb{E}[U|A=a,H].$$

Thus, the second term of Equation (3.20) is given by

$$\begin{aligned}
&m_1(\widetilde{v};\beta_0)\Big\{D(P_0)(\widetilde{o})-\Psi(P_0)(\widetilde{v})\Big\}+m_1(\widetilde{v};\beta_0)\Big\{\Psi(P_0)(\widetilde{v})-m(\widetilde{v};\beta_0)\Big\} \\
=\quad& m_1(\widetilde{v};\beta_0)\Big\{D(P_0)(\widetilde{o})-m(\widetilde{v};\beta_0)\Big\}.
\end{aligned}$$

Finally, a candidate for the efficient influence function for the parameter β_0 is given by

$$\psi(P_0)(O) = -C(P_0)^{-1} m_1(V;\beta_0)\Big\{D(P_0)(O) - m(V;\beta_0)\Big\},$$

where

$$D(P_0)(O) = \frac{I\{A=a\}}{\mathbb{P}(A=a|H)}\left[U - \mathbb{E}[U|A=a,H]\right] + \mathbb{E}[U|A=a,H]$$
$$C(P_0) = \mathbb{E}\left[m_2(V;\beta_0)\Big\{D(P_0)(O) - m(V;\beta_0\Big\} - m_1(V;\beta_0)^{\otimes 2}\right].$$

Note that in the expression for $C(P_0)$, we can replace $\Psi(P_0)(V)$ with $D(P_0)(O)$ since

$$\mathbb{E}\Big[D(P_0)(O)\Big|V\Big] = \Psi(P_0)(V).$$

This result inspires a least square type estimator for β_0. An equivalent formulation to Equation (3.18) is given by

$$\beta_0 = \arg\min_{\beta} \mathbb{E}\left[\Big(D(P_0)(O) - m(V;\beta)\Big)^2\right]$$

because

$$\mathbb{E}\left[m_1(V;\beta)\Big(D(P_0)(O) - m(V;\beta)\Big)\right]$$
$$= \mathbb{E}\left[m_1(V;\beta)\Big(\Psi(P)(V) - m(V;\beta)\Big)\right].$$

Hence, a least square type estimator for β_0 is

$$\hat{\beta} = \beta(\hat{P}_n) = \arg\min_{\beta} P_n\left[\Big(D(\hat{P}_n)(O) - m(V;\beta)\Big)^2\right],$$

for which it holds that

$$P_n\psi(\hat{P}_n)(0) = P_n\left\{-C(\hat{P}_n)^{-1} m_1(V;\hat{\beta})\Big[D(\hat{P}_n)(O) - m(V;\hat{\beta})\Big]\right\} = 0.$$

To prove that the least square estimator has influence function $\psi(P_0)(O)$, we need to show that the empirical process remainder term vanishes asymptotically, i.e., that

$$\sqrt{n}(P_n - P)\left\{\psi(\hat{P}_n)(O) - \psi(P)(O)\right\} = o_P(1),$$

and that the second order remainder term $R(\hat{P}_n, P)$ is $o_P(n^{-1/2})$. The second order remainder term is defined as

$$R(\hat{P}_n, P_0) = P_0\psi(\hat{P}_n)(O) + \Big\{\beta(\hat{P}) - \beta_0\Big\}.$$

We strongly suspect that the second order remainder is bounded by the product of the nuisance function $L_2(P_0)$ errors

$$\|Q_0(H, A) - Q(H, A)\|_{2,P_0} \times \|g_0(H, A) - g(H, A)\|_{2,P_0}.$$

However, proving this is not straightforward for a general smooth model m, and we will defer this task to future research. In conclusion, for the best smoothly parameterized least square approximation of the conditional average treatment effect or the blip function, it is possible to calculate confidence intervals for each parameter via a plug-in estimator of the efficient influence function. While this feature is not currently available in the `polle` package, we have plans to incorporate it in a future version.

Chapter 4

Reefer repair and maintenance

Maersk owns a large fleet of more than 2.5 million dry containers and 300,000 reefers used primarily to transport chilled or frozen perishable cargo. Reefers are considerably more expensive to acquire than dry containers, and with an intended lifetime between 12 and 20 years, the fleet of reefers represents a substantial long-term investment for Maersk. Maersk even produces their reefers under the brand MCI[1], see Figure 4.1.

Containers, in general, have a high wear and tear level, and as a result, Maersk spends more than $175 million annually repairing and maintaining the

[1] https://www.mcicontainers.com/

Figure 4.1: MCI Starcool reefers. The cooling fan, compressor, user display, and electronics box are clearly visible on each reefer.

fleet. Even minor improvements to the existing repair and maintenance policy can lead to significant cost savings for the business. Because of this potential, we consider asset maintenance and repairs an excellent case to illustrate the policy learning methods developed in this work, especially considering the long lifespan of reefers. The analysis and the challenges we encountered during this process formed the foundation for Paper B. The purpose of this chapter is not to repeat the details and results of the analysis but to provide some additional domain knowledge about the operation of refrigerated containers and how we define the decision problem. We also give insights into data handling and limitations.

4.1 Domain knowledge

A typical reefer is a 40ft insulated box with a T-bar ventilation floor. The box is susceptible to damage during handling, with the door and floor being particularly vulnerable. The refrigeration unit or machine is installed at the opposite end of the door. The unit's main components are the compressor, cooling fans, and electronics box. Sensors monitor the temperature inside and outside the box, and this information is transmitted to the ship or terminal. Some units also contain equipment to control the atmosphere of the box, including moisture levels and air composition. However, units with this type of equipment are not included in the analysis.

Reefers are cleaned and inspected before they are handed over to customers and when they are retuned. If a reefer is not functioning or requires maintenance, it will be sent to a local repair shop or may even be shipped to another location for major repairs. Maersk or an independent party may own the repair shop. Depending on the repair shop's capabilities, a reefer might visit several repair shops in the same location. Each repair shop creates a standardized work order that lists each task item along with the cost of materials and labor. Each item is categorized as being associated with the box or the refrigeration unit. Pictures are also included to document the list of proposed tasks. Some routine tasks, such as cleanings are automatically approved. However, the remaining work orders must be approved by a regional equipment manager who follows operational guidelines. These guidelines include age, region, and mode-specific box and refrigeration unit cost limits. The "mode" describes whether the container is laden with cargo or empty. Emergency repairs, even onboard ships, also occur frequently to save the often valuable cargo. However, our analysis will not focus on emergency repairs as part of the decision process.

A status code captures the last known status of the work order (created, edited, deleted, in progress, paid, etc.). While the status code can easily capture the approval of the work order, it is not immediately clear whether a work order has been deleted by the repair shop for technical reasons or

if the equipment manager rejected it. We spent a considerable amount of work categorizing appended comments made by the equipment manager to deduce whether the given work orders were, in fact, rejected. Another issue we encountered was that work orders split between multiple repair shops were frequently reviewed simultaneously by the equipment manager. Still, the work orders were not recorded as being linked in the system. Thus, based on the location, the creation time stamps, and the status code time stamps, we merged the work orders together to form a single entity.

4.2 Defining the problem

The approval or rejection of work orders over time forms the basis for formulating the decision process that we want to optimize. However, we only want to focus on strategically important work orders, excluding automatically approved ones. A pragmatic way to achieve this is by only considering work orders with a box or refrigeration unit cost estimate above a sufficiently high threshold. From the beginning of the project, our primary focus has been to optimize the long-term utility effects of an alternative work order policy. However, due to the gradual introduction of the operational system recording work orders in 2003 and 2004, it has not been possible to select a cohort of similar reefers for which all repairs were recorded as work orders during the complete intended lifetime of around 20 years. Instead, we decided to select a cohort of reefers produced in 2000 or 2001, which were still in active use after 5 years in the fleet. Each of these reefers were then followed for 11.5 years, resulting in a cohort of 17,883 reefers with 71,668 associated high-cost work orders.

The follow-up period, denoted as $[0, T]$, is fixed for each reefer, while the number and timing of (high-cost) work orders within the follow-up period varies. Utilizing the notation introduced in Chapter 3, Figure 4.2 illustrates how data is structured over the follow-up period. The action to approve or reject individual work orders is represented by binary variables A_k. For convenience, state variables S_k are a combination of the variables X_k and rewards U_k, which we will discuss later.

The minimal set of state variables X_k required to ensure sequential randomization, as defined in Definition (2), consists of the variables that reflect the information available to the equipment manager at the decision time point. This information includes task items, cost estimates, and reefer specifications. Furthermore, we know that the equipment manager does not review past repairs during this process. Hence, a reasonable assumption is that the action probability model is time-homogeneous:

$$g_{0,k}(H_k, A_k) = g_0(X_k, A_k),$$

for some probability function g_0. It is important to note that this does not

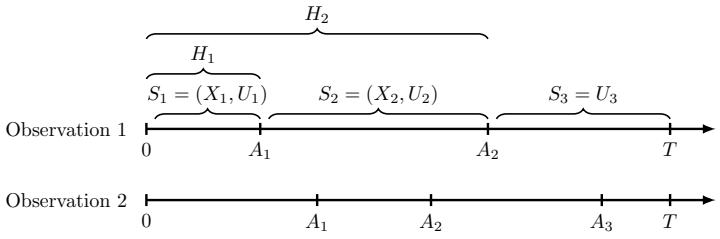


Figure 4.2: Data structure for the work orders within the follow-up period.

imply that the entire process is Markov. A Markov assumption regarding the action and state space would likely only be valid if we could accurately measure the reefer's condition. Given the existing data quality, this is not the case, meaning that the Q -functions still need to be stage-dependent. In practice, to reduce complexity, the Q-functions will not use the complete history as input; instead, we rely on a summary of the history. If this simplification introduces bias to the Q-functions, the consistency of the doubly robust policy learner and evaluation will still be guaranteed by the consistency of the g-function.

As described, the equipment managers largely adhere to regional guidelines when approving work orders. Therefore, we anticipate severe positivity violation issues for any policy learning procedure. The histogram of the cross-fitted g-function values in Figure 4.3 indeed confirms this anticipation. The propensities are heavily skewed towards 0 and 1, emphasizing the necessity for estimators that can handle positivity violations, as discussed in Section 3.5. This will also limit the potential gain of any realistic policy learner.

While the state variables discussed thus far are sufficient for identifying the optimal realistic policy, we can enhance the efficiency of the estimators and the effectiveness of the policy by incorporating variables that are predictive of future rewards. To achieve this, we include summary variables that capture the extent of reefer usage and the amount of repairs conducted.

The last missing component for the formulation of the policy learning problem is to define a suitable utility measure. The costs associated with keeping the reefer in working condition should obviously be an input to the utility measure. However, minimizing costs alone will lead to an optimal policy that rejects every work order, which is not sensible. We also need to consider the usage of each reefer during the follow-up period. Various usage metrics, such as the number of days in active service, the running time of the refrigeration unit, the number of cargo loads, and ship movements, could be considered. Ultimately, we decided to count the number of times the reefer was loaded/moved with cargo, distinguishing between whether the refrigeration unit was in use or not. This count would form the other basis of our utility measure.

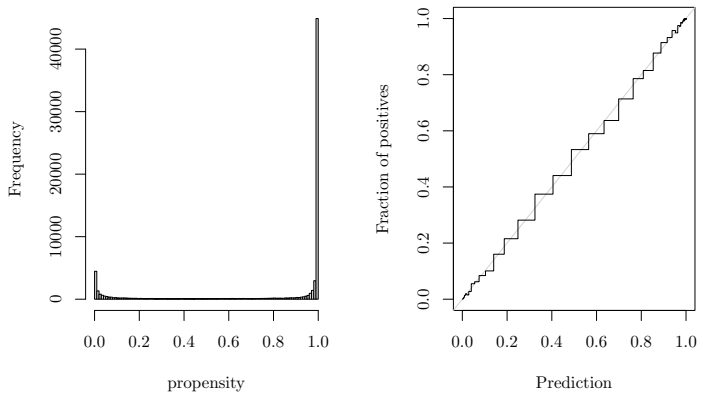


Figure 4.3: Histogram and calibration plot of the cross-fitted g-function values.

A natural way to combine the cost and number of moves into a single utility measure is to calculate the cost per move. The downside of this measure is that it prevents us from easily accumulating and adjusting the utility contributions over time. Hence, we opted for an additive structure, assigning a dollar equivalent to each type of move, which enables us to calculate the profit associated with each reefer. In addition to the operational costs and usage-related metrics, we must consider other factors. For instance, the sales or scrap price of the reefer is a vital piece of information to include. Similarly, if a reefer is still in active service at the end of the follow-up period, we must include a cash premium to account for its continued value.

There are plenty of opportunities in the future for developing the utility measure to reflect the priorities of the business even better. The stakeholders presented several ideas for improving it further. They suggested including the repositioning cost of damaged containers and adding a premium for bringing a reefer into working condition in high-demand locations, proportional to the lost profit of being unable to service the customers. Additionally, they recommended incorporating a penalty whenever cargo is lost due to a failing refrigeration unit.

4.3 Summary of findings

As suggested by the simulation study of Paper B, mimicking the extent of positivity violations seen in the actual application, even under positivity protection as described in Section 3.5, doubly robust policy learners may still suffer from the variability of inverse probability weights compared to stan-

dard Q-learning, leading to a loss in performance. However, we find that even Q-learning benefits from positivity protection by limiting the degree of extrapolation. Similarly, doubly robust evaluation of any policy learner benefits greatly from positivity protection. In the actual application, cross-fitted doubly robust policy evaluation found that realistic Q-learning with a positivity protection level of $\alpha = 0.01$ showed a significant value gain of \$287 per reefer over the follow-up period. With 300,000 reefers, this amounts to a potential saving of \$86 million.

Chapter 5

Treatment Effect Among Responders

5.1 Early response indication

For actions or treatments where the outcome is observed after a long duration of time, as is often the case in survival analysis, it is highly valuable if we can get an early indication, using a biomarker for example, of whether the patient responds to the treatment. If the patient does not respond we would be inclined to switch treatment. Similarly, in a business context, we could be interested in the effect of a type of advertisement (treatment) among the customers who were actually exposed to the ad. The question is, how do we effectively compare treatments in cases where it is possible to switch treatment based on a response indicator?

Manuscript Paper C studies the average treatment effect among responders in a survival setup under right censoring. In this setting, an observation is represented by

$$O = (H, A, D, \tilde{T}, \Delta),$$

where D is a post-randomization or post-action binary response indicator, $\tilde{T} = \min(T, C)$ for a time-to-event outcome T and censoring time C, and $\Delta = I\{T < C\}$. As before A denotes a binary treatment variable, but in this setting we assume that the treatment is completely randomized as would be the case in a clinical trial. Let $\delta = \mathbb{P}(A = 1)$ denote the randomization probability. For a given endpoint τ, the usual average treatment effect is given by

$$\mathbb{P}(T^1 \leq \tau) - \mathbb{P}(T^0 \leq \tau) = \mathbb{P}(T \leq \tau | A = 1) - \mathbb{P}(T \leq \tau | A = 0).$$

Under the assumption of independent censoring, this treatment effect can easily be identified from the observed data. However, as D represents an early indication of whether the treatment is a (complete) failure, patient dropout and the time-to-event outcome of are likely confounded by the response indicator. Of course, the treatment itself as well as other baseline variables may

also be informative for both the censoring time and the event. Thus it is more reasonable to assume that

$$T \perp C|(H, A, D),$$

which also allows us to identify the average treatment effect. The question is, how do we adjust for the response indicator? Simply conditioning on D would result in a loss of causal interpretability due to selection bias. This is because D itself is influenced by the treatment. To address this issue, we consider the principal stratum of treatment responders, defined as the set $\{D^1 = 1\}$. In a randomized trial, the potential reponse indicator D^1 is independent of the observed treatment A and therefore acts like a baseline variable. This observation leads to the definition of the average treatment effect among treatment responders as follows:

$$\mathbb{P}(T^1 \leq \tau|D^1 = 1) - \mathbb{P}(T^0 \leq \tau|D^1 = 1).$$

This causal target parameter is not directly identifiable due to the cross-world conditional probability $\mathbb{P}(T^0 \leq \tau|D^1 = 1)$. Therefore, additional structural assumptions are needed. As implied by the name of the response indicator, we assume that non-responders do not experience a treatment effect. Specifically, we assume that the average causal effect is zero among the treatment non-responders, i.e., that

$$\mathbb{P}(T^1 \leq \tau|D^1 = 0) - \mathbb{P}(T^0 \leq \tau|D^1 = 0) = 0.$$

Under the above so-called stochastic exclusion restriction and the usual structural assumptions, by the law of total probability, we see that

$$\begin{aligned} &\mathbb{P}(T^1 \leq \tau|D^1 = 1) - \mathbb{P}(T^0 \leq \tau|D^1 = 1) \\ &= \frac{\mathbb{P}(T \leq \tau|A = 1) - \mathbb{P}(T \leq \tau|A = 0)}{\mathbb{P}(D = 1|A = 1)}. \end{aligned}$$

This target parameter has been studied before by [9, 10], suggesting a simple non-parametric plug-in estimator depending on the Kaplan-Meier estimator for the treatment effect. Consistency of this estimator relies on completely independent censoring for each treatment group. The authors also suggest utilizing baseline covariates via a Cox-model. However, the very important considerations mentioned above regarding conditionally independent right censoring, as presented in Paper C, are a novel addition to the literature. The construction of the associated efficient non-parametric estimator in Paper C is an equally important contribution resulting in an attractive robust alternative to the Cox-model. Inference for the estimator is achieved via the associated influence function.

5.2 Optimal policy among responders

A natural extension to the treatment effect among responders, given our focus on optimal policies, is to investigate whether we can sensibly define an optimal policy among responders. For simplicity, we return to the usual setup where the outcome of interest is a continuous variable U. For a given unrestricted policy d, the average potential policy value is defined as follows:

$$\mathbb{E}[U^d] = \mathbb{E}[d(H)\left\{U^1 - U^0\right\}] + \mathbb{E}[U^0].$$

To avoid the need to deal with the reference value $\mathbb{E}[U^0]$, we will instead focus on the policy advantage defined as $\mathbb{E}[U^d] - \mathbb{E}[U^0]$. Conditioning on the principal stratum of treatment responders $\{D^1 = 1\}$ is no longer meaningful in the context of a policy that may select either treatment. Instead, we define the policy responders as the dynamic principal stratum given by $\{D^d = 1\}$. The policy advantage among policy responders is then defined as

$$\mathbb{E}[U^d - U^0 | D^d = 1],$$

and the optimal policy among policy responders is simply

$$\arg\max_{d \in \mathcal{D}} \mathbb{E}[U^d - U^0 | D^d = 1].$$

If the conditional average treatment effect among policy non-responders is zero, i.e., that

$$\mathbb{E}[U^d - U^0 | D^d = 0, H] = 0,$$

then the policy advantage among policy responders is identified as

$$\frac{\mathbb{E}[d(H)\mathbb{E}[Q_0(H,1) - Q_0(H,0)]]}{\mathbb{E}\big[d(H)\mathbb{E}[D|A=1,H] + (1-d(H))\mathbb{E}[D|A=0,H]\big]}.$$

If every subject responds to both treatments, we immediately recognize the optimal policy as the usual threshold policy with plug-in of the blip function. To the best of our knowledge, the formulation of the optimal policy among policy responders as stated above has not been studied in the literature. Therefore, this topic presents an exciting avenue for future research.

Chapter 6

Discussion

Policy learning is truly a vast research field across many disciplines, and thus this thesis should definitely not be seen as an exhaustive review of the topic. Rather, the project has focused on bridging the gap between the latest statistical policy learning methodologies and their practical application in the industry.

6.1 Key contributions

Linked to research objective 1, the `polle` R package Paper A offers a flexible and unified framework for conducting doubly robust policy learning and evaluation. This framework is built on a straightforward package architecture centered around three core functions: `policy_data`, `policy_learn`, and `policy_eval`. The package is user-friendly, featuring comprehensive documentation and illustrative examples. Nuisance model specifications require minimal user input, and the package automates the management of cross-fitting procedures. A noteworthy advantage of the package lies in the decomposition of policy learning and evaluation procedures, making it easy to compare competing learners in a consistent manner.

Doubly robust policy learning, as presented in this thesis, has not previously found application in industrial maintenance problems. As per research objective 2, this initiative culminated in the estimation of an improved long-term maintenance policy for Maersk reefers, resulting in a significant value increment of $287 per reefer (Paper B). In comparison to the conventional Markov decision problem formulation, our approach delivers outcomes that are causally interpretable while relying on a minimal set of structural assumptions.

The reefer maintenance application highlighted challenges related to practical positivity violations, as described in research objective 3. Within the

statistical policy learning literature, we found that methods for handling positivity violations are underdeveloped. With an increased focus on doubly robust scores, which directly utilize inverse probability weights, protecting the policy learning and evaluation process against positivity violations becomes of utmost importance. We advocate for a simple yet effective restriction on the policy learning procedure, recommending alternative actions only if they exceed a predefined probability threshold. The `polle` package introduces a novel implementation of realistic doubly robust policy learning in line with this approach.

An early indication of whether a treatment or action has the desired effect is highly valuable for outcomes with long durations, as often seen in survival analysis. Under an exclusion assumption, we can identify the treatment effect among the principal stratum of treatment responders. However, important considerations regarding right censoring, depending on the post-randomization variable itself, has been missing from the literature. In Paper C, we successfully construct an efficient estimator for the treatment effect among responders under right censoring, leveraging informative baseline covariates and adjusting for the post randomization indicator. This estimator holds high relevance for the design of future treatment switching policy designs.

6.2 Future work and research

As mentioned, many important policy learning topics have not been covered in this thesis. We intend to continue developing the `polle` package, and we hope to incorporate substantial additional functionality in the future. The following is a summary of some of the topics we consider important for future work:

Doubly robust policy learners also exist for Markov decision problem formulations [22, 25]. An obvious extension of `polle` would be to include these learners as well. The policy data object of `polle` already has the functionality to handle Markov data in long data table format.

From a practical perspective, incorporating solutions for (right) censoring [61, 52, 5] and missing data [46, 31] would be highly valuable. The event variable included in the policy data object of `polle` was designed from the beginning to be used for dealing with censored data.

Adding variable importance measures for smoothly parameterized blip functions, as discussed in Section 3.7, is a logical extension for the `polle` package. This implementation would improve the interpretation of the learned policy. This topic is closely related to interpretable policies, as explored by [59].

As mentioned briefly in Section 3.3, subgroup analysis is closely intertwined with policy learning. This analysis is particularly valuable, especially in clinical trials where the active treatment is compared to a placebo treatment. The value of subgroup analysis can also extend to business problems, for example, when creating targeted marketing campaigns. Implementing subgroup analysis in the single-stage scenario is straightforward within the `polle` framework. However, for the multi-stage case, it necessitates more careful considerations.

Throughout this work, we have concentrated on discrete action sets. Extending the `polle` package to accommodate continuous actions or treatments would undoubtedly increase its usability. However, dealing with non-parametric evaluation of the learned policy in such cases becomes substantially more complex [26, 27, 3].

Another related challenge revolves around optimizing 'when-to-treat' policies [36]. In the context of maintenance problems, this is often referred to as predictive maintenance. Unlike our application, where we do not intervene in the timing between decision points, the problem of optimizing dynamic interventions on a general counting process is considerably more intricate. To our knowledge, a solution to this problem has only been convincingly formulated for equidistant decision time points.

So far, we have only considered deterministic policies. However, in most cases, we aim to implement decision processes that continue to explore, with the goal of further optimization. Consequently, the estimated policy should not be directly implemented but instead be subject to stochastic modifications. This topic is extensively addressed in the reinforcement learning literature, where the objective is to strike the right balance between policy greediness and exploration level [47]. Regarding trial design, [35] discusses sequentially randomized trials, including sample size and power calculations. The concept of data-adaptive designs, where the implemented policy is continuously updated, is explored in works like [14, 7].

Finally, from a methodological perspective, we want to mention the development of the optimal policy among responders as discussed in Section 5.2. These types of policy estimates can really help in designing effective sequentially randomized trials or experiments where the treatment switches based on post-randomization response indicators.

The success of policy learning in a company like Maersk does not only rely on the development of better methods, but it critically depends on the business committing to documenting decision processes, conducting experimentation, collecting data, and defining utility measures that reflect the priorities of the business. All these efforts require that Maersk adapts its company culture and continues to support efforts like this project. We sincerely believe that this work contributes to transforming Maersk into being truly data-driven in all aspects of operations and business. We also hope that the academic research

community realizes just how important causal inference and statistical policy learning are for making a real change in the private sector. Collaborations like this project benefit all parties and can help push the field forward.

Bibliography

[1] Eva Ascarza. Retention futility: Targeting high-risk customers might be ineffective. *Journal of Marketing Research*, 55(1):80–98, 2018.

[2] Susan Athey. Beyond prediction: Using big data for policy problems. *Science*, 355(6324):483–485, 2017.

[3] Susan Athey and Stefan Wager. Policy learning with observational data. *Econometrica*, 89(1):133–161, 2021.

[4] Jean-Yves Audibert and Alexandre B Tsybakov. Fast learning rates for plug-in classifiers. *Annals of Statistics*, 35(2):608–633, 2007.

[5] Xiaofei Bai, Anastasios A Tsiatis, Wenbin Lu, and Rui Song. Optimal treatment regimes for survival endpoints using a locally-efficient doubly-robust estimator from a classification perspective. *Lifetime Data Analysis*, 23:585–604, 2017.

[6] Oliver Bembom and Mark J van der Laan. A practical illustration of the importance of realistic individualized treatment rules in causal inference. *Electronic Journal of Statistics*, 1:574, 2007.

[7] Aurélien Bibaut, Nathan Kallus, Maria Dimakopoulou, Antoine Chambaz, and Mark van Der Laan. Risk minimization from adaptively collected data: Guarantees for supervised and policy learning. *Advances in Neural Information Processing Systems*, 34:19261–19273, 2021.

[8] Peter J Bickel, Chris AJ Klaassen, Peter J Bickel, Ya'acov Ritov, J Klaassen, Jon A Wellner, and YA'Acov Ritov. *Efficient and adaptive estimation for semiparametric models*, volume 4. Johns Hopkins University Press Baltimore, 1993.

[9] Björn Bornkamp and Georgina Bermann. Estimating the treatment effect in a subgroup defined by an early post-baseline biomarker measurement in randomized clinical trials with time-to-event endpoint. *Statistics in Biopharmaceutical Research*, 2019.

[10] Björn Bornkamp, Kaspar Rufibach, Jianchang Lin, Yi Liu, Devan V Mehrotra, Satrajit Roychoudhury, Heinz Schmidli, Yue Shentu, and Marcel Wolbers. Principal stratum strategy: potential role in drug development. *Pharmaceutical Statistics*, 20(4):737–751, 2021.

[11] Andreas Buja, Lawrence Brown, Arun Kumar Kuchibhotla, Richard Berk, Edward George, and Linda Zhao. Models as approximations ii. *Statistical Science*, 34(4):545–565, 2019.

[12] Bibhas Chakraborty, Eric B Laber, and Ying-Qi Zhao. Inference about the expected performance of a data-driven dynamic treatment regime. *Clinical Trials*, 11(4):408–417, 2014.

[13] Bibhas Chakraborty and EE Moodie. *Statistical methods for dynamic treatment regimes*. Springer-Verlag, 2013.

[14] Antoine Chambaz, Wenjing Zheng, and Mark J van der Laan. Targeted sequential design for targeted learning inference of the optimal treatment rule and its mean reward. *Annals of Statistics*, 45(6):2537, 2017.

[15] Aaron Fisher and Edward H Kennedy. Visually communicating and teaching intuition for influence functions. *The American Statistician*, 75(2):162–172, 2021.

[16] Yair Goldberg and Michael R Kosorok. Q-learning with censored data. *Annals of Statistics*, 40(1):529, 2012.

[17] Oliver Hines, Oliver Dukes, Karla Diaz-Ordaz, and Stijn Vansteelandt. Demystifying statistical learning based on efficient influence functions. *The American Statistician*, 76(3):292–304, 2022.

[18] Keisuke Hirano and Jack R Porter. Asymptotics for statistical treatment rules. *Econometrica*, 77(5):1683–1701, 2009.

[19] Keisuke Hirano and Jack R Porter. Impossibility results for nondifferentiable functionals. *Econometrica*, 80(4):1769–1790, 2012.

[20] Xinyang Huang, Yair Goldberg, and Jin Xu. Multicategory individualized treatment regime using outcome weighted learning. *Biometrics*, 75(4):1216–1227, 2019.

[21] Hidehiko Ichimura and Whitney K Newey. The influence function of semiparametric estimators. *Quantitative Economics*, 13(1):29–61, 2022.

[22] Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 652–661. PMLR, 2016.

[23] Nathan Kallus. Recursive partitioning for personalization using observational data. In *International Conference on Machine Learning*, pages 1789–1798. PMLR, 2017.

[24] Nathan Kallus. More efficient policy learning via optimal retargeting. *Journal of the American Statistical Association*, 116(534):646–658, 2021.

[25] Nathan Kallus and Masatoshi Uehara. Double reinforcement learning for efficient off-policy evaluation in markov decision processes. *The Journal of Machine Learning Research*, 21(1):6742–6804, 2020.

[26] Nathan Kallus and Angela Zhou. Policy evaluation and optimization with continuous treatments. In *International conference on artificial intelligence and statistics*, pages 1243–1251. PMLR, 2018.

[27] Edward H Kennedy. Towards optimal doubly robust estimation of heterogeneous causal effects. *arXiv preprint arXiv:2004.14497*, 2020.

[28] Edward H Kennedy, Sivaraman Balakrishnan, James M Robins, and Larry Wasserman. Minimax rates for heterogeneous causal effect estimation. *arXiv preprint arXiv:2203.00837*, 2022.

[29] Toru Kitagawa and Aleksey Tetenov. Who should be treated? empirical welfare maximization methods for treatment choice. *Econometrica*, 86(2):591–616, 2018.

[30] Michael R Kosorok and Eric B Laber. Precision medicine. *Annual Review of Statistics and its Application*, 6:263–286, 2019.

[31] Michael R Kosorok and Erica EM Moodie. *Adaptive treatment strategies in practice: planning trials and analyzing data for personalized medicine*. SIAM, 2015.

[32] Ying Liu, Yuanjia Wang, Michael R Kosorok, Yingqi Zhao, and Donglin Zeng. Augmented outcome-weighted learning for estimating optimal dynamic treatment regimens. *Statistics in Medicine*, 37(26):3776–3788, 2018.

[33] Alex Luedtke and Incheoul Chung. One-step estimation of differentiable hilbert-valued parameters. *arXiv preprint arXiv:2303.16711*, 2023.

[34] Alexander R Luedtke and Mark J Van Der Laan. Statistical inference for the mean outcome under a possibly non-unique optimal treatment strategy. *Annals of Statistics*, 44(2):713, 2016.

[35] Susan A Murphy. An experimental design for the development of adaptive treatment strategies. *Statistics in Medicine*, 24(10):1455–1481, 2005.

[36] Xinkun Nie, Emma Brunskill, and Stefan Wager. Learning when-to-treat policies. *Journal of the American Statistical Association*, 116(533):392–409, 2021.

[37] Xinkun Nie and Stefan Wager. Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*, 108(2):299–319, 2021.

[38] Maya L Petersen, Kristin E Porter, Susan Gruber, Yue Wang, and Mark J Van Der Laan. Diagnosing and responding to violations in the positivity assumption. *Statistical Methods in Medical Research*, 21(1):31–54, 2012.

[39] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[40] Min Qian and Susan A Murphy. Performance guarantees for individualized treatment rules. *Annals of Statistics*, 39(2):1180, 2011.

[41] Thomas S Richardson and James M Robins. Single world intervention graphs (swigs): A unification of the counterfactual and graphical approaches to causality. *Center for the Statistics and the Social Sciences, University of Washington Series. Working Paper*, 128(30):2013, 2013.

[42] James M Robins. Optimal structural nested models for optimal sequential decisions. In *Proceedings of the second seattle Symposium in Biostatistics*, pages 189–326. Springer, 2004.

[43] Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688, 1974.

[44] Donald B Rubin. Randomization analysis of experimental data: The fisher randomization test comment. *Journal of the American Statistical Association*, 75(371):591–593, 1980.

[45] Vira Semenova and Victor Chernozhukov. Debiased machine learning of conditional average treatment effects and other causal functions. *The Econometrics Journal*, 24(2):264–289, 2021.

[46] Susan M Shortreed, Eric Laber, T Scott Stroup, Joelle Pineau, and Susan A Murphy. A multiple imputation strategy for sequential multiple assignment randomized trials. *Statistics in Medicine*, 33(24):4202–4214, 2014.

[47] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[48] Anastasios A Tsiatis, Marie Davidian, Shannon T Holloway, and Eric B Laber. *Dynamic treatment regimes: Statistical methods for precision medicine*. Chapman and Hall/CRC, 2019.

[49] Masatoshi Uehara, Chengchun Shi, and Nathan Kallus. A review of off-policy evaluation in reinforcement learning. *arXiv preprint arXiv:2212.06355*, 2022.

[50] Masatoshi Uehara and Wen Sun. Pessimistic model-based offline reinforcement learning under partial coverage. *arXiv preprint arXiv:2107.06226*, 2021.

[51] Mark J Van der Laan. Statistical inference for variable importance. *The International Journal of Biostatistics*, 2(1), 2006.

[52] Mark J van der Laan and Alexander R Luedtke. Targeted learning of an optimal dynamic treatment, and statistical inference for its mean outcome. *UC Berkeley Division of Biostatistics Working Paper Series*, 2014.

[53] Mark J Van der Laan and James M Robins. *Unified methods for censored longitudinal data and causality*, volume 5. Springer-Verlag, 2003.

[54] Mark J Van der Laan, Sherri Rose, et al. *Targeted learning: causal inference for observational and experimental data*, volume 4. Springer, 2011.

[55] Aad W Van der Vaart. *Asymptotic Statistics*, volume 3. Cambridge university press, 2000.

[56] Stijn Vansteelandt and Oliver Dukes. Assumption-lean inference for generalised linear model parameters. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(3):657–685, 2022.

[57] Baqun Zhang, Anastasios A Tsiatis, Eric B Laber, and Marie Davidian. A robust method for estimating optimal treatment regimes. *Biometrics*, 68(4):1010–1018, 2012.

[58] Chong Zhang, Jingxiang Chen, Haoda Fu, Xuanyao He, Ying-Qi Zhao, and Yufeng Liu. Multicategory outcome weighted margin-based learning for estimating individualized treatment rules. *Statistica Sinica*, 30:1857, 2020.

[59] Yichi Zhang, Eric B Laber, Marie Davidian, and Anastasios A Tsiatis. Interpretable dynamic treatment regimes. *Journal of the American Statistical Association*, 113(524):1541–1549, 2018.

[60] Yingqi Zhao, Donglin Zeng, A John Rush, and Michael R Kosorok. Estimating individualized treatment rules using outcome weighted learning. *Journal of the American Statistical Association*, 107(499):1106–1118, 2012.

[61] Yufan Zhao, Donglin Zeng, Mark A Socinski, and Michael R Kosorok. Reinforcement learning strategies for clinical trials in nonsmall cell lung cancer. *Biometrics*, 67(4):1422–1433, 2011.

[62] Zhengyuan Zhou, Susan Athey, and Stefan Wager. Offline multi-action policy learning: Generalization and optimization. arxiv preprint arxiv. *arXiv preprint arXiv:1810.04778*, 2018.

Appendix I

Data-adaptive policy value empirical process remainder

Let $\hat{d}_n(H)$ denote a policy estimator for which we assume that

$$P_0|\hat{d}_n(H) - d'(H)| = o_{P_0}(1),$$

for some policy d'. Furthermore, conditional on $\hat{Q}_n$ and $\hat{g}_n$, assume that

$$\begin{aligned} \|\hat{Q}_n(H,A) - Q_0(H,A)\|_{2,P_0} &= o_{P_0}(1) \\ \|\hat{g}_n(H,A) - g_0(H,A)\|_{2,P_0} &= o_{P_0}(1). \end{aligned} \tag{I.1}$$

Under these conditions, we want to show that the data-adaptive policy value empirical process remainder is $o_{P_0}(n^{-1/2})$, i.e., that

$$\{P_n - P_0\}\left\{\psi(\hat{g}_n, \hat{Q}_n, \hat{d}_n)(X) - \psi(g_0, Q_0, d')(X)\right\} = o_{P_0}(n^{-1/2}). \tag{I.2}$$

We start by noting that

$$\{P_n - P_0\}\left\{\Psi(\hat{g}_n, \hat{Q}_n, \hat{d}_n) - \Psi(g_0, Q_0, d')\right\} = 0.$$

Thus, a sufficient condition for (I.2) is that $\hat{Q}_n$, $\hat{g}_n$, and $\hat{d}_n$ falls in a Donsker class with probability approaching one, and that conditional on $\hat{g}_n$, $\hat{Q}_n$, and $\hat{d}_n$

$$\begin{aligned} &\left\|\hat{Q}_n(H, \hat{d}_n(H)) - Q_0(H, d'(H))\right\|_{2,P_0} = o_{P_0}(1) \\ &\left\|\frac{I\{A = \hat{d}_n(H)\}}{\hat{g}_n(H,A)}\left\{U - \hat{Q}_n(H,A)\right\} - \frac{I\{A = d'(H)\}}{g_0(H,A)}\{U - Q_0(H,A)\}\right\|_{2,P_0} = o_{P_0}(1). \end{aligned}$$

Firstly,

$$\begin{aligned} &\|\hat{Q}_n(H, \hat{d}_n(H)) - Q_0(H, d'(H))\|_{2,P_0} \\ \leq\ &\|\hat{Q}_n(H, \hat{d}_n(H)) - \hat{Q}_n(H, d'(H))\|_{2,P_0} \\ &+\ \|\hat{Q}_n(H, d'(H)) - Q_0(H, d'(H))\|_{2,P_0} \end{aligned}$$

The last of the above terms is bounded by Line (I.1) under positivity. Assuming that $\hat{Q}_n$ is bounded by a constant C as in Theorem 5, we also see that

$$\begin{aligned}
&\left\|\hat{Q}_n(H,\hat{d}_n) - \hat{Q}_n(H,d'(H))\right\|_{2,P_0} \\
\leq &\left\|I\{\hat{d}_n(H) \neq d'(H)\}\left\{\hat{Q}_n(H,\hat{d}_n(H)) - \hat{Q}_n(H,d'(H))\right\}\right\|_{2,P_0} \\
\leq &\, 2C\left\|I\{\hat{d}_n(H) \neq d'(H)\}\right\|_{2,P_0} \\
= &\, o_{P_0}(1)
\end{aligned}$$

Secondly,

$$\begin{aligned}
&\left\|\frac{I\{A=\hat{d}_n(H)\}}{\hat{g}_n(H,A)}(U-\hat{Q}_n(H,A)) - \frac{I\{A=d'(H)\}}{g_0(H,A)}(U-Q_0(H,A))\right\|_{2,P_0} \\
\leq &\epsilon^{-2}\left\|g_0(H,A)I\{A=\hat{d}_n(H)\}[U-\hat{Q}_n(H,A)] - \hat{g}_n(H,A)I\{A=d'(H)\}[U-Q_0(H,A)]\right\|_{2,P_0} \\
\leq &\epsilon^{-2}\left\|g_0(H,A)\left\{I\{A=\hat{d}_n(H)\}[U-\hat{Q}_n(H,A)] - I\{A=d'(H)\}[U-Q_0(H,A)]\right\}\right\|_{2,P_0} \\
+&\epsilon^{-2}\left\|\{g_0(H,A)-\hat{g}_n(H,A)\}\, I\{A=d'(H)\}[U-Q_0(H,A)]\right\|_{2,P_0}.
\end{aligned}$$

Now

$$\begin{aligned}
&\left\|\{g_0(H,A)-\hat{g}_n(H,A)\}\, I\{A=d'(H)\}[U-Q_0(H,A)]\right\|_{2,P_0} \\
\leq &\, 2C\left\|\{g_0(H,A)-\hat{g}_n(H,A)\}\right\|_{2,P_0} \\
= &\, o_{P_0}(1),
\end{aligned}$$

and

$$\begin{aligned}
&\left\|g_0(H,A)\left\{I\{A=\hat{d}_n(H)\}[U-\hat{Q}_n(H,A)] - I\{A=d'(H)\}[U-Q_0(H,A)]\right\}\right\|_{P_0,2} \\
\leq &\left\|I\{A=\hat{d}_n(H)\}[U-\hat{Q}_n(H,A)] - I\{A=d'(H)\}[U-Q_0(H,A)]\right\|_{P_0,2} \\
\leq &\left\|I\{A=\hat{d}_n(H)\}[U-\hat{Q}_n(H,A)] - I\{A=\hat{d}_n(H)\}[U-Q_0(H,A)]\right\|_{P_0,2} \\
&+\left\|I\{A=\hat{d}_n(H)\}[U-Q_0(H,A)] - I\{A=d'(H)\}[U-Q_0(H,A)]\right\|_{P_0,2} \\
\leq &\left\|[U-\hat{Q}_n(H,A)] - [U-Q_0(H,A)]\right\|_{P_0,2} \\
&+2C\left\|I\{A=\hat{d}_n(H)\} - I\{A=d'(H)\}\right\|_{P_0,2} \\
= &\, o_{P_0}(1).
\end{aligned}$$

Paper A

Policy Learning with the polle package

Authors:
Andreas Nordland & Klaus K. Holst

Publication details:
Submitted to the Journal of Statistical Software

Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

Policy Learning with the polle package

Andreas Nordland
Section of Biostatistics
University of Copenhagen

Klaus Kähler Holst
Global Data Analytics
A.P. Moeller-Maersk

Abstract

The R package **polle** is a unifying framework for learning and evaluating finite stage policies based on observational data. The package implements a collection of existing and novel methods for causal policy learning including doubly robust restricted Q-learning, policy tree learning, and outcome weighted learning. The package deals with (near) positivity violations by only considering realistic policies. Highly flexible machine learning methods can be used to estimate the nuisance components and valid inference for the policy value is ensured via cross-fitting. The library is built up around a simple syntax with four main functions `policy_data()`, `policy_def()`, `policy_learn()`, and `policy_eval()` used to specify the data structure, define user-specified policies, specify policy learning methods and evaluate (learned) policies. The functionality of the package is illustrated via extensive reproducible examples.

Keywords: policy learning, dynamic treatment regimes, semiparametric inference, double machine learning, R.

1. Introduction

Sequential decision problems arise in various fields. Important examples include deciding on treatment assignments in a medical application, defining equipment maintenance strategies in a military or industrial setting, or determining a sales strategy in a commercial context. Policy learning seeks to identify sequential decision strategies from observational data and to quantify the effect of implementing such a strategy using causal inference techniques. While the theoretical field has progressed substantially during the last decade based on advances in semiparametric methods, there has been a large gap in terms of generic implementations of these methods being available to practitioners.

The R package **polle** (Nordland and Holst 2022) is a unifying framework for learning optimal policies/dynamic treatment regimes from historical data based on cross-fitted doubly robust

loss functions for finite horizon problems with discrete action sets. Within this scope, the package unifies available methods from other R packages and introduces previously unavailable methods. The performance of the methods can then easily be evaluated, compared and applied to new data. As a unique feature, the package also handles a stochastic number of decision stages. In addition, the package deals with issues related to learning optimal policies from observed data under (near) positivity violations by considering *realistic policies*.

The core concept of **polle** is to use doubly robust scores/double machine learning developed from semiparametric theory when estimating the value of a policy (Robins 1986; Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey, and Robins 2018). These scores are also used to construct a doubly robust loss function for the optimal policy value (Tsiatis, Davidian, Holloway, and Laber 2019). The resulting loss function is the basis for policy value search within a restricted class of policies such as policy trees (Athey and Wager 2021). Transformations of the value loss function leads to a range of other loss functions and methods such as doubly robust restricted Q-learning (Luedtke and van der Laan 2016) and outcome weighted learning based on support vector machines (Zhang, Tsiatis, Davidian, Zhang, and Laber 2012; Zhao, Zeng, Rush, and Kosorok 2012). As is customary within targeted learning and double machine learning, our policy evaluation and policy learning methods apply cross-fitting schemes, which allow for inference under weak conditions even when nuisance parameters are learned from highly flexible machine learning methods.

Sequential policy learning is closely related to estimating heterogeneous causal effects via the conditional average treatment effect, see (Kennedy 2020; Semenova and Chernozhukov 2021) for recent overviews of the field and some of the challenges regarding inference and generic error bounds. Other notable mentions include (Künzel, Sekhon, Bickel, and Yu 2019) and (Athey, Tibshirani, and Wager 2019). Variable importance measures formulated as projections are also closely related to policy learning, see (Van der Laan 2006).

The **polle** R package joins a collection of other packages available on CRAN, `https://CRAN.R-project.org/view=CausalInference`. Other packages which should be highlighted include **DynTxRegime** (Holloway, Laber, Linn, Zhang, Davidian, and Tsiatis 2022) which provides methods for estimating policies including interactive Q-learning, outcome weighted learning, and value search. However, most of the methods are only implemented for single stage problems and the package has no cross-fitting methods for consistent policy evaluation. The **polle** package wraps efficient augmentation and relaxation learning and residual weighted learning from the **DynTxRegime** package. The package **policytree** (Sverdrup, Kanodia, Zhou, Athey, and Wager 2020, 2022) is an implementation of single stage policy tree value search based on doubly robust scores. The **polle** package wraps this functionality and extends it to a stochastic number of stages. Lastly, the R package **DTRlearn2** (Chen, Liu, Zeng, and Wang 2020) implements outcome weighted learning in a fixed number of stages. The **polle** package also wraps this functionality.

Beyond R, the Python package **EconML** (Battocchi, Dillon, Hei, Lewis, Oka, Oprescu, and Syrgkanis 2019) implements a wide range of learners for the conditional average treatment effect including doubly robust estimators (equivalent to doubly robust Q-learning as formulated in **polle**), double machine learning estimators, and orthogonal random forests. The package also implements policy trees and forests. To our knowledge, **EconML** does not contain methods for cross-fitted policy evaluation. For multi-stage decision problems, the package only considers G-estimation based on specific Markov decision process structural equation models, see (Lewis and Syrgkanis 2020).

The available methods for policy learning in proprietary software are still very limited. A SAS macro denoted `PROC QLEARN` performs standard Q-learning (Ertefaie, Almirall, Huang, Dziak, Wagner, and Murphy 2012). In stata methhods are limited to estimating average treatment effects and potential outcome means with the `teffects` function (StataCorp 2021).

In Section 2 we introduce the most important concepts of doubly robust policy learning in a simple single-stage setting. In doing so, we avoid the cumbersome notation needed for the general sequential setup as presented in Section 3. In Section 4 we give an overview of the package syntax and describe the main functions of the package. Section 5 contains four reproducible examples based on simulated data covering all aspects of the package. In Section 6 we present a complete analysis of a data set investigating the treatment effect of a literacy intervention. Finally, in Section 7 we summarize the functionality of **polle** and discuss limitations and future developments.

2. Concepts

In a randomized trial investigating the average treatment effect of two competing treatments we should ask ourselves whether the treatment effect is heterogeneous or not, i.e., whether the subjects respond differently to the treatments depending on their age, sex, disease history, etc. If so, is it possible to learn a treatment policy from the observed data that will have a greater expected outcome than any of the individual treatments?

For simplicity, we consider a single-stage problem, where each subject receives a completely randomized treatment at a single time point. Let A denote the treatment variable with two levels $A \in \{0, 1\}$, and let U denote the measured utility outcome. The average treatment effect is a causal parameter which can be formulated via potential outcomes (Rubin 1974; Hernán and Robins 2020). We let U^a denote the potential utility had we forced the subject to receive treatment $A = a$, and we refer to $\mathsf{E}[U^a]$ as the value of the given treatment. The average treatment effect is now defined as the difference in value, $\mathsf{E}[U^1 - U^0]$, and due to complete randomization, the effect is identified as $\mathsf{E}[U|A = 1] - \mathsf{E}[U|A = 0]$. The effect is easily estimated based on a sample of N iid observations $O = (A, U)$. Without additional information, treatment $A = 1$ is recommended if the estimated effect is positive and vice versa.

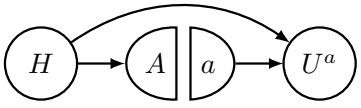


Figure 1: Single world intervention graph illustrating confounding via the history.

Suppose now that we also collect a set of baseline covariates $H \in \mathcal{H}$ for each subject, and that treatment randomization depends on this history by design. The treatment probability model is then given by a known function

$$g_0(h, a) = \mathsf{P}(A = a|H = h). \tag{1}$$

If the trial has a sensible design we will also know that $g_0(H, a) > 0$ almost surely for $a \in \{0, 1\}$, which is commonly referred to as the positivity condition. Due to confounding,

the average treatment effect will no longer be identified by the mean utility in each treatment group, see Figure 1. However, it is possible to show that

$$\mathsf{E}\left[\frac{I\{A=a\}}{g_0(H,a)}U\right] = \mathsf{E}[U^a]. \tag{2}$$

This equality inspires an inverse probability weighting (Horvitz-Thompson) estimator for the value of each treatment group, see (Horvitz and Thompson 1952). In an observational study, the treatment probability function g_0 might not be known a priori. In that case, we instead use some appropriate estimate g_N.

Alternatively, the treatment value is identified as

$$\mathsf{E}[\mathsf{E}[U|A=a,H]] = \mathsf{E}[U^a], \tag{3}$$

where we define the quality function as $Q_0(h,a) = \mathsf{E}[U|A=a,H=h]$. Usually, the Q-function is not known a priori and will need to be estimated. The fit Q_N is then used to construct an outcome regression estimate of the value in each treatment group, see (Robins 1986).

If the treatment values are heterogeneous across the collected history it is possible that one group of subjects benefit from treament $A=1$ and that another group of subjects benefit from treament $A=0$. The researcher may even have a candidate treatment policy $d : \mathcal{H} \to \{0,1\}$ that he believes will improve the value. Let U^d denote the potential utility had we forced the subject to be treated in accordance to policy d. The policy value $E[U^d]$ can then be estimated using (2) or (3) or a combination of the two. Define the doubly robust policy score as

$$Z(d,g,Q)(O) = Q(H,d(H)) + \frac{I\{A=d(H)\}}{g(H,A)}\left(U - Q(H,A)\right). \tag{4}$$

If either $g = g_0$ or $Q = Q_0$ it holds that

$$\mathsf{E}\left[Z(d,g,Q)(O)\right] = \mathsf{E}[U^d]. \tag{5}$$

The associated empirical plug-in estimator is said to be doubly robust. Furthermore, it is possible to show that the estimator is asymptotically efficient, if the nuisance models (g_N and Q_N) are correctly specified, see (Van der Laan and Robins 2003). Specifically, the centralized score

$$Z(d,g,Q)(O) - \mathsf{E}\left[Z(d,g,Q)(O)\right]$$

is the efficient influence function from which we can derive the asymptoic distributions via central limit theorem arguments. For a recent review of influence functions, see (Hines, Dukes, Diaz-Ordaz, and Vansteelandt 2022).

By applying nuisance model cross-fitting (/sample splitting) in combination with the doubly robust score, the nuisance models can be estimated using flexible machine learning methods without causing asymptotic bias (Chernozhukov *et al.* 2018). In Section 3.2 we present the estimating procedure of the policy value in detail and generalize it to multi-category policies over multiple stages.

In many situations the aim of the researcher is not just to evaluate a given policy, but to learn the optimal policy from the data. The optimal treatment policy d_0 is defined as the policy for which it holds that $\mathsf{E}[U^d] \leq \mathsf{E}[U^{d_0}]$ for all other policies d. Thus, a direct approach to policy learning is to use (5) as a loss function:

$$d_N = \arg\min_{d\in\mathcal{D}} \sum_{i=1}^{N} \tilde{L}(d)(g_N, Q_n)(O_i) = \arg\min_{d\in\mathcal{D}} (-1) \sum_{i=1}^{N} Z(d, g_N, Q_n)(O_i).$$

In pratice, the complexity of the class of candidate policies $\mathcal{D}$ is bounded for the search to be viable. Examples include threshold policies and policy trees, see (Athey and Wager 2021). In the first part of Section 3.3 we present this methodology in detail and generalize it to multiple stages.

A key and rather intuitive result is that the optimal policy is also identified as

$$d_0(h) = \arg\max_{a\in\{0,1\}} \mathsf{E}[U^a|H = h] = \arg\max_{a\in\{0,1\}} Q_0(H, a). \tag{6}$$

This result motivates Q-learning which rely on estimating the Q-function. The fitted Q-function is then plugged into (6) to get the associated estimated policy. Although the implementation of standard Q-learning is straightforward, estimation of the associated policy value requires parametric convergence rates of the estimated Q-function (van der Laan and Luedtke 2014; Semenova and Chernozhukov 2021). Ignoring this by using too flexible machine learning methods makes it impossible to put any standard bounds on the policy performance. On the other hand, misspecification of the model will introduce bias and as a consequence lead to poor performing policies. To deal with this problem we advocate the use of *restricted doubly robust Q-learning*. Let $V \in \mathcal{V}$ be a subset or a function of the history H. Let $d^V : \mathcal{V} \to \{0, 1\}$ denote a V-restricted policy. Finally, let $\mathcal{D}^V$ denote the class of V-restricted policies. The optimal V-restricted policy is simply defined as the policy d_0^V for which it holds that $\mathsf{E}[U^{d^V}] \leq \mathsf{E}[U^{d_0^V}]$ for all $d^V \in \mathcal{D}^V$. Similarly as above, the optimal V-restricted policy is given by

$$d_0^V(v) = \arg\max_{a\in\{0,1\}} \mathsf{E}\left[U^a|V = v\right] = \arg\max_{a\in\{0,1\}} QV_0(v, a),$$

see (Luedtke and van der Laan 2016). The QV-function is now identified in two ways:

$$\mathsf{E}\left[\frac{I\{A = a\}}{g_0(H, a)} U \middle| V\right] = \mathsf{E}\left[Q_0(H, a)|V\right] = QV_0(V, a).$$

Again, the nuisance models can combined to create a doubly robust expression for the optimal V-restricted policy. Define $Z(a, g, Q)$ as $Z(d, g, Q)$ from line (4) under the static policy $d(H) = a$. If either $g = g_0$ or $Q = Q_0$ it holds that

$$\mathsf{E}\left[Z(a, g, Q)(O)|V\right] = QV_0(V, a).$$

This result directly inspires a doubly robust regression type estimator for the QV-function. Specifically, we let QV_N denote the function with the lowest empirical mean squared error loss:

$$\begin{aligned} QV_N(\cdot, a) &= \arg\min_{QV} \sum_{i=1}^{N} L(QV)(g_N, Q_N)(O_i) \\ &= \arg\min_{QV} \sum_{i=1}^{N} \Big(Z\left(a, g_N, Q_N\right)(O_i) - QV\left(V_i, a\right)\Big)^2, \end{aligned}$$

where we let the class of candidate QV-functions have bounded complexity, e.g., it must be a member of a Donsker class such as the class of smooth parametric models (Luedtke and Chambaz 2020). This QV-function formulated as a projection in a mean squared error sense remedies the shortcomings of standard Q-learning and also improves the interpretability of the resulting policy. We present doubly roubust Q-learning (DRQ-learning) in detail in the second part of Section 3.3 where we also generalize it to multiple stages.

The final concept that we want to introduce for now is realistic policy learning. Positivity violations, see (1), or even near positivity violations is a concern for both policy learning and evaluation (Petersen, Porter, Gruber, Wang, and Van Der Laan 2012). If we in some stratum of the history do not observe both treatments it is impossible to learn the optimal policy in the given stratum without strong structural assumptions. Thus we introduce the set of (estimated) reaslistic actions at level α:

$$D_N^\alpha(h) = \{a \in \mathcal{A} : g_N(h, a) > \alpha\}.$$

DRQ-learning can then easily be adapted to the set of realistic actions as follows

$$d_N(h) = \arg\max_{a \in D_N^\alpha(h)} QV_N(v, a).$$

In the next section we formalize all of the above concepts and generalize them to multiple stages.

3. Setup and methods

3.1. General multi-stage setup

Let $K \geq 1$ denote a fixed number of stages. Let $B \in \mathcal{B}$ denote the baseline covariates. For a finite set $\mathcal{A}$, let $A_k \in \mathcal{A}$ denote the decision or action at stage $k \in \{1, \dots, K\}$. For $k \in \{1, \dots, K+1\}$, let $S_k \in \mathcal{S}$ denote the state at stage k. The trajectory for an observation can be written as

$$O = (B, S_1, A_1, S_2, A_2, \dots, S_K, A_K, S_{K+1}),$$

as illustrated in Figure 2. Usually, we will assume to have a sample of N iid observations indexed as $\{O_i\}_{i \in 1 \dots, N}$. For $k \in \{1, \dots, K+1\}$, let $\overline{S}_k = (S_1, \dots, S_k)$, $\overline{A}_k = (A_1, \dots, A_k)$ and $H_k = (B, \overline{S}_k, \overline{A}_{k-1}) \in \mathcal{H}_k$ define the history where $A_0 = A_{K+1} = \emptyset$. Using the implied ordering, the density of the data can be expressed as

$$p_0(O) = p_0(B) \left[\prod_{k=1}^{K} p_{0,k}(A_k | H_k) \right] \left[\prod_{k=1}^{K+1} p_{0,k}(S_k | H_{k-1}, A_{k-1}) \right]. \tag{7}$$

For convenience, let $S_k = (X_k, U_k)$, where $U_k \in \mathbb{R}$ is the kth reward, and X_k is a state covariate/variable for $k \in \{1, \dots, K\}$ and $X_{K+1} = \emptyset$. The utility is the sum of the rewards

$$U = \sum_{k=1}^{K+1} U_k.$$

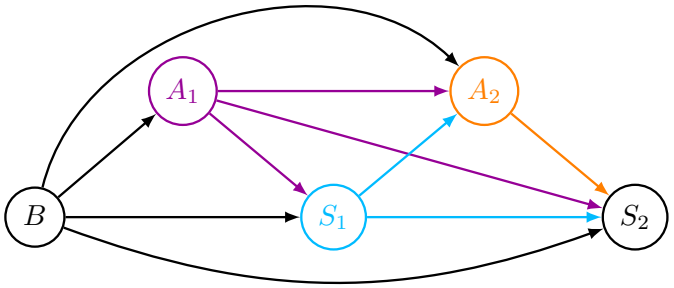


Figure 2: Graph for the observational data with two stages. B is a baseline covariate, A_1, A_2 are the two decisions at stages 1 and 2, and S_1, S_2 are the state variables. From each of the state variables, a reward can be derived, and the sum of these defines the utility of the decisions.

3.2. Policy value estimation

A policy is a set of rules $d = (d_1, ..., d_K)$, $d_k : \mathcal{H}_k \mapsto \mathcal{A}$ assigning an action in each stage. Let $D_{0,k}(h_k) \subseteq \mathcal{A}$ denote the feasible set of decicions at stage k for history h_k under P_0, i.e.,

$$D_{0,k}(h_k) = \{a_k \in \mathcal{A} : p_{0,k}(a_k|h_k) > 0\}.$$

Define the class of feasible policies $\mathcal{D}_0$ as all sets of rules satisfying $d_k(h_k) \in D_{0,k}(h_k)$.
For a feasible policy d, let P_0^d be the distribution with density

$$p_0^d(O) = p_0(B) \left[\prod_{k=1}^{K} I\{A_k = d_k(H_k)\}\right] \left[\prod_{k=1}^{K+1} p_{0,k}(S_k|H_{k-1}, A_{k-1})\right]. \tag{8}$$

Let O^d denote the data with distribution given by (8), which is identified from the observed data. Define the value of the policy as

$$\theta_0^d = \mathsf{E}[U^d].$$

Under consistency and sequential randomization the above value will have a causal interpretation as the mean utility under an intervention given by the feasible policy.

The value of a feasible policy d can explicitly be stated via the Q-functions recursively defined as

$$Q_{0,K}(h_K, a_K) = \mathsf{E}[U \mid H_K = h_K, A_K = a_K]$$

$$Q_{0,k}^{\underline{d}_{k+1}}(h_k, a_k) = \mathsf{E}\left[Q_{0,k+1}^{\underline{d}_{k+2}}(H_{k+1}, d_{k+1}(H_{k+1})) \mid H_k = h_k, A_k = a_k\right], \quad k \in \{1, \ldots, K-1\}$$

where $\underline{d}_k = (d_k, ..., d_K)$. It is possible to show that the target parameter is identified as

$$\theta_0^d = \mathsf{E}[Q_{0,1}^d(H_1, d_1(H_1)].$$

The recursive structure of the Q-functions directly inspires a sequential regression procedure resulting in an estimate $Q^d_{N,1}$, based on N iid observations. The value can then be estimated as the empirical mean of $Q^d_{N,1}(H_1, d_1(H_1))$.

The value of a feasible policy d can also be stated via the g-functions defined as

$$g_{0,k}(h_k, a_k) = p_{0,k}(a_k \mid h_k),$$

for $k \in \{1, \ldots, K\}$. Again, it is possible to show that

$$\theta^d_0 = \mathsf{E}\left[\left(\prod_{k=1}^{K} \frac{I\{A_k = d_k(H_k)\}}{g_{0,k}(H_k, A_k)}\right) U\right].$$

Given regression estimates $g_{N,k}$, the value can now be estimated as the weighted empirical mean of the observed utilities.

Finally, it is possible to combine the two estimation approaches. Define the doubly robust score at stage k as

$$\begin{aligned} &Z_k(\underline{d}_k, g, Q^{\underline{d}_{k+1}})(O) \\ &\quad = Q_k^{d_{k+1}}(H_k, d_k(H_k)) \\ &\quad + \sum_{r=k}^{K} \left\{ \prod_{j=k}^{r} \frac{I\{A_j = d_j(H_j)\}}{g_j(H_j, A_j)} \right\} \left\{ Q_{r+1}^{\underline{d}_{r+2}}(H_{r+1}, d_{r+1}(H_{r+1})) - Q_r^{\underline{d}_{r+1}}(H_r, d_r(H_r)) \right\}, \end{aligned} \quad (9)$$

where $Q_{K+1}(H_{K+1}, d_{K+1}(H_{K+1})) = U$. It is possible to show that $\mathsf{E}[Z_1(d, g, Q^d)(O)] = \theta^d_0$ if either $g = g_0$ or $Q^d = Q^d_0$, see for example Tsiatis *et al.* (2019). This result directly inspires a doubly robust moment type estimator of the policy value, see Algorithm 1.

Algorithm 1: Cross-fitted doubly robust estimator of θ^d_0

input : Data set with iid observations $\mathcal{O} = (O_1, \ldots, O_N)$
Feasible policy d
Action probability regression procedure $\hat{g}$
Outcome regression procedure $\hat{Q}^d$

output: Value estimate θ^d_N
Variance estimate Σ^d_N

$\{\mathcal{O}_1, \ldots, \mathcal{O}_M\} =$ `M-folds` $(\mathcal{O})$
foreach $m \in \{1, \ldots, M\}$ **do**
  $g_m = \hat{g}(\mathcal{O} \setminus \mathcal{O}_m)$
  $Q^d_m = \hat{Q}^d(\mathcal{O} \setminus \mathcal{O}_m)$
  $\mathcal{Z}_{1,m} = \{Z_1(d, g_m, Q^d_m)(O) : O \in \mathcal{O}_m\}$
end
$\theta^d_N = N^{-1} \sum_{m=1}^{M} \sum_{Z \in \mathcal{Z}_{1,m}} Z$
$\Sigma^d_N = N^{-1} \sum_{m=1}^{M} \sum_{Z \in \mathcal{Z}_{1,m}} (Z - \theta^d_N)^2$

It is well known that $\psi_0^d(O) = Z_1(d, g_0, Q_0^d)(O) - \theta_0^d$ is the efficient influence function/curve for the policy value. We assume that the absolute utility is bounded and that $g_{k,0}(H_k, A_K) > \epsilon$ almost surely for some $\epsilon > 0$. Let $\|g\|_{P,2} = \max_{j\in\{1,...,K\}} \|g_j\|_{P,2}$ and $\|Q^d\|_{P,2} = \max_{j\in\{1,...,K\}} \|Q_j^{\underline{d}_{j+1}}\|_{P,2}$. If, with probability converging to one, $g_{k,m}(H_k, A_k) > \epsilon$ and

$$\begin{aligned} \|g_m - g_0\|_{P_0,2} &= o_{P_0}(1) \\ \|Q_m^d - Q_0^d\|_{P_0,2} &= o_{P_0}(1) \\ \|g_m - g_0\|_{P_0,2} \times \|Q_m^d - Q_0^d\|_{P_0,2} &= o_{P_0}(N^{-1/2}), \end{aligned}$$

then

$$N^{1/2}(\theta_N^d - \theta_0^d) = N^{-1/2} \sum_{i=1}^{N} \psi_0^d(O_i) + o_{P_0}(1).$$

Thus Σ_N^d from Algorithm 1 is a good estimate of the asymptotic variance of the value estimate if the nuisance models $\hat{g}$ and $\hat{Q}^d$ are correctly specified. It is important to note that the convergence rate conditions are relatively weak. For example, Q_0^d and g_0 may be estimated at rate $o(N^{-1/4})$, which is much lower than a parametric rate of order $o(N^{-1/2})$. This justifies the use of adaptive and regularized nuisance models, see Chernozhukov *et al.* (2018).

3.3. Policy learning

The main objective of **polle** is to learn the optimal policy from data. Specifically, we want to learn the optimal policy within a restricted class of policies such that the learned policy is simpler to understand and easier to implement. We start by defining the optimal policy within a restricted class of policies that only depends on a subset of the observed history. The following result is a generalization of van der Laan and Luedtke (2014).

Let V_k be a function (or subset) of H_k. A V-restricted policy is a set of rules $d^V = (d_1^V, ..., d_K^V)$, $d_k^V : \overline{\mathcal{A}}_{k-1} \times \mathcal{V}_k \mapsto \mathcal{A}$. Let $\mathcal{D}^V$ denote the class of V-restricted policies. Under positivity, i.e., $D_{0,k}(H_k) = \mathcal{A}$ almost surely, the V-optimal policy is defined as

$$d_0^V = \arg\max_{d\in\mathcal{D}^V} \mathsf{E}[U^d].$$

The following theorem specifies the V-optimal policy. A proof for the two-stage case can be found in Appendix A.

Theorem 3.1:
Under positivity, for any $a = (a_1, ..., a_K)$ and policy d define

$$QV_{0,K}(\overline{a}_{K-1}, v_K, a_K) = \mathsf{E}[U^{\overline{a}_K} | V_K^{\overline{a}_{K-1}} = v_K], \tag{10}$$

$$QV_{0,k}^{\underline{d}_{k+1}}(\overline{a}_{k-1}, v_k, a_k) = \mathsf{E}[U^{\overline{a}_k, \underline{d}_{k+1}} | V_k^{\overline{a}_{k-1}} = v_k] \quad k \in \{1, ..., K-1\}. \tag{11}$$

If

$$\mathsf{E}[U^a | V_1, ..., V_k^{\overline{a}_{k-1}}] = \mathsf{E}[U^a | V_k^{\overline{a}_{k-1}}], \quad k \in \{1, ..., K\}, \tag{12}$$

then the V-optimal policy d_0^V is recursively given by

$$d_{0,K}^V(\overline{a}_{K-1}, v_K) = \arg\max_{a_K} QV_{0,K}(\overline{a}_{K-1}, v_K, a_K),$$
$$d_{0,k}^V(\overline{a}_{k-1}, v_k) = \arg\max_{a_k} QV_{0,k}^{\underline{d}_{0,k+1}^V}(\overline{a}_{k-1}, v_k, a_k) \quad k \in \{1, ..., K-1\}.$$

If for all $k \in \{1, ..., K\}$ and $r < k$, $V_r^{\overline{a}_{r-1}}$ is a function of $V_k^{\overline{a}_{k-1}}$ then (12) holds by construction. Finally, note that only future rewards affects the optimal decision at stage k since

$$\begin{aligned}
&\arg\max_{a_k} QV_{0,k}^{\underline{d}_{0,k+1}^V}(\overline{a}_{k-1}, v_k, a_k) \\
&\qquad = \arg\max_{a_k} \mathsf{E}\left[U_1 + ... + U_k^{\overline{a}_{k-1}} + U_{k+1}^{\overline{a}_k} + ... + U_{K+1}^{\overline{a}_k, \underline{d}_{0,k+1}^V} | V_k^{\overline{a}_{k-1}} = v_k\right] \\
&\qquad = \arg\max_{a_k} \mathsf{E}\left[U_{k+1}^{\overline{a}_k} + ... + U_{K+1}^{\overline{a}_k, \underline{d}_{0,k+1}^V} | V_k^{\overline{a}_{k-1}} = v_k\right].
\end{aligned}$$

The basis for learning the V-optimal policy is to construct an observed data loss function L which identifies d_0^V, i.e., construct a function L for which $\mathsf{E}[L(d)(O)]$ is minimized in d_0^V. Various loss functions inspires different algorithms for estimating the V-optimal policy. In the following, we present three different loss functions, a value, quality and classification loss function.

Value search

For the final stage K consider the loss function $\tilde{L}_K(d_K)(g_K, Q_K)(O)$ in d_K given by

$$\begin{aligned}
-\tilde{L}_K(d_K)(g_K, Q_K)(O) &= Z_K(d_K, g, Q)(O) \\
&= Q_K(H_k, d_K(H_k)) + \frac{I\{A_K = d_K(H_k)\}}{g_K(H_K, A_K)} \{U - Q_K(H_K, A_K)\}
\end{aligned}$$

If either $Q_K = Q_{0,K}$ or $g_K = g_{0,K}$, then

$$\mathsf{E}\left[\tilde{L}_K(d_K)(g_K, Q_K)(O)\right] = -\mathsf{E}\left[U^{d_K}\right].$$

Thus, for a V-restricted policy

$$\mathsf{E}\left[\tilde{L}_K(d_K^V)(g_K, Q_K)(O)\right] = -\mathsf{E}\left[QV_{0,K}\left(\overline{A}_{K-1}, V_K, d_K^V(\overline{A}_{K-1}, V_K)\right)\right],$$

meaning that over the class of V-restricted policies $\mathcal{D}_K^V$ the expected loss is minimized in $d_{0,K}^V$ by Theorem 3.1.

For stage $k \in \{1, ..., K-1\}$ consider the loss function $\tilde{L}_k(d_k)(\underline{d}_{k+1}, g, Q^{\underline{d}_{k+1}})(O)$ in d_k given by

$$-\tilde{L}_k(d_k)(\underline{d}_{k+1}, g, Q^{\underline{d}_{k+1}})(O) = Z_k([d_k, \underline{d}_{k+1}], g, Q^{\underline{d}_{k+1}})(O).$$

If either $Q^{\underline{d}_{k+1}} = Q_0^{\underline{d}_{k+1}}$ or $g = g_0$, then

$$E\left[\tilde{L}_k(d_k)(\underline{d}_{k+1}, g, Q^{\underline{d}_{k+1}})(O)\right] = -E\left[U^{d_k, \underline{d}_{k+1}}\right],$$

and for a V-restricted policy at stage k it holds that

$$E\left[\tilde{L}_k(d_k^V)(\underline{d}_{k+1}, g, Q^{\underline{d}_{k+1}})(O)\right] = -E\left[QV_{0,k}^{\underline{d}_{k+1}}(\overline{A}_{k-1}, V_k, d_k^V(\overline{A}_{k-1}, V_k)\right]$$

Thus, given $\underline{d}_{0,k+1}^V$, the above expected loss over $\mathcal{D}_k^V$ is minimized in $d_{0,k}^V$ by Theorem 3.1.

The constructed loss function directly inspires sequential/recursive value search, see Algorithm 2. Similar to Algorithm 1, the algorithm utilizes cross-fitted values of the nuisance models at each step. However, it is important to note that the Q-models are not truly cross-fitted (except for the last stage), because the fitted policy at a given stage depends on the fitted policy at later stages. A nested cross-fitting scheme would be required to make the folds (used to fit the Q-models) independent.

Algorithm 2: Sequential Value Search

input : Data set with iid observations $\mathcal{O} = (O_1, \ldots, O_N)$
Class of V-restricted policies $\mathcal{D}^V$
Function class minimization procedure $\hat{F}$
Action probability regression procedure $\hat{g}$
Outcome regression procedure $\hat{Q} = \{\hat{Q}_1, ..., \hat{Q}_K\}$
output: V-restricted optimal policy estimate d_N^V

$\{\mathcal{O}_1, \ldots, \mathcal{O}_L\} =$ `L-folds` $(\mathcal{O})$
foreach $l \in \{1, \ldots, L\}$ **do**
  $g_l = \hat{g}(\mathcal{O} \setminus \mathcal{O}_l)$
end
for $k = K$ **to** 1 **do**
  foreach $l \in \{1, \ldots, L\}$ **do**
    $Q_{l,k}^{d_{N,k+1}^V} = \hat{Q}_k^{d_{N,k+1}^V}\left(\left\{H_k, A_k, Q_{k+1,l}^{d_{N,k+2}^V}\left(H_{k+1}, d_{N,k+1}^V(H_{k+1})\right) : O \in \mathcal{O} \setminus \mathcal{O}_l\right\}\right)$
  end
  $d_{N,k}^V = \hat{F}_{d_k^V \in \mathcal{D}^V}\left(\sum_{l=1}^L \sum_{O \in \mathcal{O}_l} \tilde{L}(d_k^V)(\underline{d}_{k+1,N}^V, g_l, Q_l^{\underline{d}_{k+1,N}^V})(O)\right)$
end

Algorithm 2 requires a suitable function class minimization procedure $\hat{F}$ imitating $\arg\min_{d_k^V \in \mathcal{D}^V}\{\cdot\}$ at every stage. The R package **policytree**, see Sverdrup *et al.* (2020), implements such a minimization procedure where the class of policies is given by decision trees. See Zhou, Athey, and Wager (2018) for theoretical results related to this implementation.

Quality learning

Under positivity, for any $a = (a_1, ..., a_K)$ and any policy d, let $Z_k([a_k, \underline{d}_{k+1}], g, Q^{\underline{d}_{k+1}})(O)$ for $k \in \{1, ..., K\}$ be given by (9) with d_k replaced by the static policy $a_k \in \mathcal{A}$.

For the final stage K, define $QV_{0,K}(a_K)(\overline{a}_{K-1}, v_K) = QV_{0,K}(\overline{a}_{K-1}, v_K, a_K)$ from equation

(10). If $g_K = g_{0,K}$ or $Q_K = Q_{0,K}$ then

$$\mathsf{E}\left[Z_K(a_K, g, Q)(O)|\overline{A}_{K-1}, V_K\right] = QV_{0,K}(a_K)(\overline{A}_{K-1}, V_K),$$

and valid loss function for $QV_{0,K}(a_K)$ over functions $QV_K : \overline{\mathcal{A}}_{K-1} \times \mathcal{V}_K \mapsto \mathcal{A}$ is given by

$$L_K(QV_K)(a_K, g, Q)(O) = \left(Z_K(a_K, g, Q)(O) - QV_K(\overline{A}_{K-1}, V_K)\right)^2.$$

Hence, any regression type estimator which minimizes the (empirical) mean squared error can be used to estimate $QV_{0,K}(a_K)$. This can be repeated for every $a_K \in \mathcal{A}$. By Theorem 3.1, the V-optimal policy $d^V_{0,K}$ is then identified as $\arg\max_{a_K \in \mathcal{A}} QV_{0,K}(a_K)$.

For $k \in \{1, ..., K-1\}$, let $QV^{\underline{d}_{k+1}}_{0,k}(a_K)(\overline{a}_{k-1}, v_k) = QV^{\underline{d}_{k+1}}_{0,k}(\overline{a}_{k-1}, v_k, a_k)$ from equation (11). If $g = g_0$ or $Q^{\underline{d}_{k+1}} = Q^{\underline{d}_{k+1}}_0$ then

$$\mathsf{E}\left[Z_k([a_k, \underline{d}_{k+1}], g, Q^{\underline{d}_{k+1}})(O)|\overline{A}_{k-1}, V_k\right] = QV^{\underline{d}_{k+1}}_{0,k}(a_k)(\overline{A}_{k-1}, V_k),$$

and a valid loss function for $QV^{\underline{d}_{k+1}}_{0,k}(a_k)$ over functions QV_k is given by

$$L_k(QV_k)(a_k, \underline{d}_{k+1}, g, Q^{\underline{d}_{k+1}})(O) = \left(Z_k([a_k, \underline{d}_{k+1}], g, Q^{\underline{d}_{k+1}})(O) - QV_k(\overline{A}_{k-1}, V_k)\right)^2.$$

Thus, given the future V-restricted optimal policy rules $\underline{d}^V_{0,k+1}$, if $g = g_0$ or $Q^{\underline{d}^V_{0,k+1}} = Q^{\underline{d}^V_{0,k+1}}_0$, then the expected loss is minimized in $QV^{\underline{d}^V_{0,k+1}}_{0,k}(a_k)$. Again, this can be repeated for each $a_k \in \mathcal{A}$ and the V-optimal policy at stage k is identified as $\arg\max_{a_k \in \mathcal{A}} QV^{\underline{d}^V_{0,k+1}}_{0,k}(a_k)$.

The constructed quality loss function directly inspires doubly robust Q-learning (DRQ-learning), see Algorithm 3. The quality loss function is a generalization of the blip loss function, see Appendix B.

Weighted classification

In this section we assume for simplicity that the actions are binary, i.e., $\mathcal{A} = \{0, 1\}$. Define the blip or Z-score difference as

$$\breve{Z}_k(\underline{d}_{k+1}, g, Q^{\underline{d}_{k+1}})(O) = \left\{Z_k([1, \underline{d}_{k+1}], g, Q^{\underline{d}_{k+1}})(O) - Z_k([0, \underline{d}_{k+1}], g, Q^{\underline{d}_{k+1}})(O)\right\}.$$

A weighted classification (0-1) loss function is now given by

$$\begin{aligned}\breve{L}_k(d_k)(\underline{d}_{k+1}, g, Q^{\underline{d}_{k+1}})(O) = &\left|\breve{Z}_k(\underline{d}_{k+1}, g, Q^{\underline{d}_{k+1}})(O)\right| \\ &\times I\Big\{d_k(H_k) \neq I\Big\{\breve{Z}_k(\underline{d}_{k+1}, g, Q^{\underline{d}_{k+1}})(O) > 0\Big\}\Big\}. \qquad (13)\end{aligned}$$

Given the future V-optimal policy rules $\underline{d}^V_{0,k+1}$, if $g = g_0$ or $Q^{\underline{d}^V_{0,k+1}} = Q^{\underline{d}^V_{0,k+1}}_0$, then it can be shown that the expected weighted classification loss function is minimized in $d^V_{0,k}$ over $\mathcal{D}^V_k$. Further details on the weighted classification loss function can be found in Appendix C.

Algorithm 3: Doubly Robust Q-learning

input : Data set with iid observations $\mathcal{O} = (O_1, \ldots, O_N)$
Action probability regression procedure $\hat{g}$
Outcome regression procedure $\hat{Q} = \{\hat{Q}_1, ..., \hat{Q}_K\}$
Outcome regression procedure $\widehat{QV} = \{\widehat{QV}_1, ..., \widehat{QV}_K\}$
output: V-restricted optimal policy estimate d_N^V

$\{\mathcal{O}_1, \ldots, \mathcal{O}_L\} =$ `L-folds` $(\mathcal{O})$
foreach $l \in \{1, \ldots, L\}$ **do**
  $g_l = \hat{g}(\mathcal{O} \setminus \mathcal{O}_l)$
end
for $k = K$ **to** 1 **do**
  foreach $l \in \{1, \ldots, L\}$ **do**
    $Q_{k,l}^{\underline{d}_{N,k+1}^V} = \hat{Q}_k \left(\left\{ H_k, A_k, Q_{k+1,l}^{\underline{d}_{N,k+2}^V} \left(H_{k+1}, d_{N,k+1}^V(H_{k+1}) \right) : O \in \mathcal{O} \setminus \mathcal{O}_l \right\} \right)$
    foreach $a_k \in \mathcal{A}$ **do**
      $\widetilde{\mathcal{O}}_{k,l}(a_k) = \left\{ \overline{A}_{k-1}, V_k, Z_k \left([a_k, \underline{d}_{N,k+1}^V], g_l, Q_l^{\underline{d}_{N,k+1}^V} \right)(O) : O \in \mathcal{O}_l \right\}$
    end
  end
  foreach $a_k \in \mathcal{A}$ **do**
    $QV_{N,k}^{\underline{d}_{N,k+1}^V}(a_k) = \widehat{QV}_k \left(\{ \widetilde{O} \in \widetilde{\mathcal{O}}_{k,l}(a_k) : l = 1, ..., L \} \right)$
  end
  $d_{N,k}^V = \arg\max_{a_k \in \mathcal{A}} QV_{N,k}^{\underline{d}_{N,k+1}^V}(a_k)$
end

It can be challenging to perform minimization of the weighted classification loss function. Thus, it is common to use a convex surrogate of the indicator function. Let $f_k : \mathcal{H}_k \mapsto \mathbb{R}$ be some action function corresponding to d_k, i.e., $d_k(H_k) = I\{f_k(H_k) > 0\}$. Then

$$\breve{L}_k(f_k)(\underline{d}_{k+1}, g, Q^{\underline{d}_{k+1}})(O) = \left|\breve{Z}_k(\underline{d}_{k+1}, g, Q^{\underline{d}_{k+1}})(O)\right| \times I\left\{f_k(H_k)\left[2I\left\{\breve{Z}_k(\underline{d}_{k+1}, g, Q^{\underline{d}_{k+1}})(O) > 0\right\} - 1\right] \leq 0\right\}$$

is equivalent to (13). Replacing $I\{x \leq 0\}$ in the above expression with a convex surrogate $\phi : \mathbb{R} \mapsto [0, \infty)$ differentiable in 0 with $\phi'(0) < 0$ yields a convex loss function given by

$$\breve{L}_k^{\phi}(f_k)(\underline{d}_{k+1}, g, Q^{d})(O) = \left|\breve{Z}_k(\underline{d}_{k+1}, g, Q^{\underline{d}_{k+1}})(O)\right| \times \phi\left(f_k(H_k)\left[2I\left\{\breve{Z}_k(\underline{d}_{k+1}, g, Q^{\underline{d}_{k+1}})(O) > 0\right\} - 1\right]\right).$$

If $g = g_0$ or $Q^{d_0^V} = Q_0^{d_0^V}$ and the non-exceptional law holds, i.e., that

$$0 < \mathsf{E}\left[\breve{Z}_k(\underline{d}_{k+1}, g, Q^{\underline{d}_{k+1}})(O)\right], \tag{14}$$

then the expected weighted surrogate loss function is minimized in $f_{0,k}^V$ over the class of V-restricted action functions and $d_{0,k}^V = I\{f_{0,k}^V > 0\}$. The above result directly inspires sequential learning of the restricted optimal policy using weighted classification methods similar to Algorithm 2.

The classification perspective was first established by Zhao *et al.* (2012) and Zhang *et al.* (2012). Various methods within this approach has been implemented in the R packages **DTRlearn2** and **DynTxRegime**, see Chen *et al.* (2020) and Holloway *et al.* (2022). Generalizations to multiple actions (more than two) has also been developed, see Zhang, Chen, Fu, He, Zhao, and Liu (2020).

3.4. Learned policy value estimation

Although the optimal (restricted) policy value is identified, existence of the associated efficient influence function is not guaranteed. Using the results of Hirano and Porter (2012), Luedtke and Van Der Laan (2016) show that the efficient influence function only exists under the non-exceptional law, i.e., that the action which minimizes the quality at each stage is almost surely unique, see line (14). For references on the non-regularity issues causing these problems, see Robins and Rotnitzky (2014) and Chakraborty and Moodie (2013). Even under the non-exceptional law, inference for the learned policy value requires an unreasonably high policy learning rate, as the Q-functions are required to be estimated at rate $o(N^{-1/2})$. Thus, we draw inference on the value of the learned policy, $\mathsf{E}\left[U^{d_N^V}\right]$, instead of the value of the true optimal policy, $\mathsf{E}\left[U^{d_0^V}\right]$, see Algorithm 4.

3.5. Stochastic number of stages

The methodology developed for a fixed number of stages can be extended to handle a stochastic number of stages assuming that the maximal number of stages is finite. The key is to

Algorithm 4: Cross-fitted doubly robust estimator of $\theta_0^{d_N}$

input : Data set with iid observations $\mathcal{O} = (O_1, \ldots, O_N)$
Policy learning procedure $\hat{d}$
Action probability regression procedure $\hat{g}$
Outcome regression procedure $\hat{Q}^d$
output: Cross-fitted value estimate θ_N^d
Cross-fitted variance estimate Σ_N^d

$\{\mathcal{O}_1, \ldots, \mathcal{O}_M\} =$ `M-folds` $(\mathcal{O})$
foreach $m \in \{1, \ldots, M\}$ **do**
  $d_m = \hat{d}(\mathcal{O} \setminus \mathcal{O}_m)$
  $g_m = \hat{g}(\mathcal{O} \setminus \mathcal{O}_m)$
  $Q_m^{d_m} = \hat{Q}^{d_m}(\mathcal{O} \setminus \mathcal{O}_m)$
  $\mathcal{Z}_{1,m} = \{Z_1(d_m, g_m, Q_m^{d_m})(O) : O \in \mathcal{O}_m\}$
end
$\theta_N^{d_N} = N^{-1} \sum_{m=1}^M \sum_{Z \in \mathcal{Z}_{1,m}} Z$
$\Sigma_N^{d_N} = N^{-1} \sum_{m=1}^M \sum_{Z \in \mathcal{Z}_{1,m}} (Z - \theta_N^{d_N})^2$

modify each observation such that every observation has the same number of stages.

Let K^* denote the stochastic number of stages bounded by a maximal number of stages K. As in Section 3.1, let B^* denote the baseline data, $(A_1^*, \ldots, A_{K^*}^*)$ denote the decisions and $(S_1^*, \ldots, S_{K^*+1}^*)$ denote the stage summaries where $S_k^* = (X_k^*, U_k^*)$ and $X_{K^*+1}^* = \emptyset$. The utility U^* is still the sum of the rewards $U^* = \sum_{k=1}^{K^*+1} U_k^*$. We assume that the distribution of the observed data is given by P_0^* composed of conditional densities $p_0^*(B^*)$, $p_{0,k}^*(A_k^* \mid H_k^*)$ for $k \in \{1, ..., K\}$ and $p_{0,k}^*(S_k^* \mid H_k^*)$ for $k \in \{1, ..., K+1\}$ such that the likelihood for an observation O^* is given by

$$p_0^*(O^*) = p_0^*(B^*) \left[\prod_{k=1}^{K^*} p_{0,k}^*(A_k^* | H_k^*) \right] \left[\prod_{k=1}^{K^*+1} p_{0,k}^*(S_k^* | H_{k-1}^*, A_{k-1}^*) \right].$$

For a feasible policy $d^* = (d_1^*, ..., d_K^*)$ the distribution P^{*d^*} is defined similar to P^d in (8). Also, the value under P^{*d^*} is defined as $\mathsf{E}\left[U^{*d}\right]$.

We now construct auxiliary data such that each observation O^* has K stages. Let $A_k = A_k^*$ for $k \leq K^*$ and $A_k = a^\dagger \in \mathcal{A}$ (for some default value $a^\dagger$) for $k > K^*$. Similarly, let $S_k = S_k^*$ for $k \leq K^* + 1$. Finally, let $X_k = \emptyset$ and $U_k = 0$ for $k > K^* + 1$ such that $U = U^*$. This construction implies a partly degenerate distribution P_0 over the maximal number of stages with density on the form given by (7), see Goldberg and Kosorok (2012). A feasible policy d associated with d^* is given by

$$d_k(H_k) = \begin{cases} a^\dagger & \text{if } X_k = \emptyset \\ d_k^*(H_k) & \text{otherwise.} \end{cases}$$

Furthermore, it holds by construction that $g_{0,k}(H_k, a^\dagger) = 1$ and $Q_{0,k}^d(H_k, a^\dagger) = U$ if $X_k = \emptyset$.

Finally, a generalization of the results in Goldberg and Kosorok (2012) yields that $\mathsf{E}\left[U^{d}\right] = \mathsf{E}\left[U^{*d^*}\right]$. Thus, the methodology developed for a fixed number of stages can be used on the augmented data.

3.6. Partial policy

It may occur that a small subset of the observations has numerous stages. Without further structural assumptions, information about these late stages will be sparse. Uncertain estimation of the Q-functions for the late stages can be avoided by considering partial policies. Let $\tilde{K} < K$ and let $\overline{d}_{\tilde{K}}$ be a given policy up till stage $\tilde{K}$. A partial (stochastic) policy is now given by $(\overline{d}_{\tilde{K}}, A_{\tilde{K}+1}, \ldots, A_K)$. By setting $Q_{0,\tilde{K}} = \mathsf{E}[U \mid H_{\tilde{K}} = h_{\tilde{K}}, A_{\tilde{K}} = a_{\tilde{K}}]$ and $Q_{0,\tilde{K}+1} = U$, the efficient influence score for the partial policy value will be equal to (9) with K replaced by $\tilde{K}$. From a practical point of view, implementation of a partial policy requires that $(g_{0,\tilde{K}+1}, \ldots, g_{0,K})$ is known (or at least well approximated).

3.7. Learning realistic policies

Positivity violations or even near positivity violations is a huge concern for policy learning based on historical data. Estimation of a valid loss function will solely rely on extrapolation of the Q-functions to decisions with little or no support in the observed data, see Petersen *et al.* (2012). To address this issue we suggest restricting the set of possible interventions based on the action probability model. For a probability threshold $\alpha > 0$, define the set of realistic actions at stage k based on the action probability model g as

$$D^{\alpha}_{g,k}(h_k) = \{a_k \in \mathcal{A} : g_k(h_k, a_k) > \alpha\}.$$

It is relatively simple to modify doubly robust Q-learning to only consider realistic policies, see Algorithm 5. On the other hand, it is harder to make the same practical modification to a given value search algorithm because the structure of the candidate function class $\mathcal{D}^V$ changes in a non-trivial way.

However, in the situation that the action set is dichotomous, the recommended action can be overruled by the alternative action, if it is deemed unrealistic.

Algorithm 5: Realistic Doubly Robust Q-learning

input : Data set with iid observations $\mathcal{O} = (O_1, \ldots, O_N)$
Action probability regression procedure $\hat{g}$
Outcome regression procedure $\hat{Q} = \{\hat{Q}_1, ..., \hat{Q}_K\}$
Outcome regression procedure $\hat{QV} = \{\hat{QV}_1, ..., \hat{QV}_K\}$
output: Realistic V-restricted optimal policy estimate d_N^V

$g_N = \hat{g}(\mathcal{O})$
$\{\mathcal{O}_1, \ldots, \mathcal{O}_L\} =$ `L-folds` $(\mathcal{O})$
foreach $l \in \{1, \ldots, L\}$ **do**
  $g_l = \hat{g}(\mathcal{O} \setminus \mathcal{O}_l)$
end
for $k = K$ **to** 1 **do**
  foreach $l \in \{1, \ldots, L\}$ **do**
    $Q_{k,l}^{\underline{d}_{N,k+1,l}^V} = \hat{Q}_k \left(\left\{ H_k, A_k, Q_{k+1,l}^{\underline{d}_{N,k+2,l}^V} \left(H_{k+1}, d_{N,k+1,l}^V(H_{k+1}) \right) : O \in \mathcal{O} \setminus \mathcal{O}_l \right\} \right)$
    foreach $a_k \in \mathcal{A}$ **do**
      $\widetilde{\mathcal{O}}_{k,l}(a_k) = \left\{ \overline{A}_{k-1}, V_k, Z_k \left([a_k, \underline{d}_{N,k+1,l}^V], g_l, Q_l^{\underline{d}_{N,k+1,l}^V} \right) (O) : O \in \mathcal{O}_l \right\}$
    end
  end
  foreach $a_k \in \mathcal{A}$ **do**
    $QV_{N,k}^{\underline{d}_{N,k+1}^V}(a_k) = \hat{QV}_k \left(\{ \widetilde{O} \in \widetilde{\mathcal{O}}_{k,l}(a_k) : l = 1, ..., L \} \right)$
  end
  foreach $l \in \{1, \ldots, L\}$ **do**
    $d_{N,k,l}^V = \underset{a_k \in D_{g_l,k}^{\alpha}}{\arg\max}\ QV_{N,k}^{\underline{d}_{N,k+1}^V}(a_k)$
  end
  $d_{N,k}^V = \underset{a_k \in D_{g_N,k}^{\alpha}}{\arg\max}\ QV_{N,k}^{\underline{d}_{N,k+1}^V}(a_k)$
end

4. Syntax and implementation details

The **polle** implementation is build up around four functions: `policy_data()`, `policy_def()`, `policy_eval()` and `policy_learn()`. Figure 3 provides an overview of how the functions relate and the main required inputs and outputs. `policy_data()` constructs a policy data

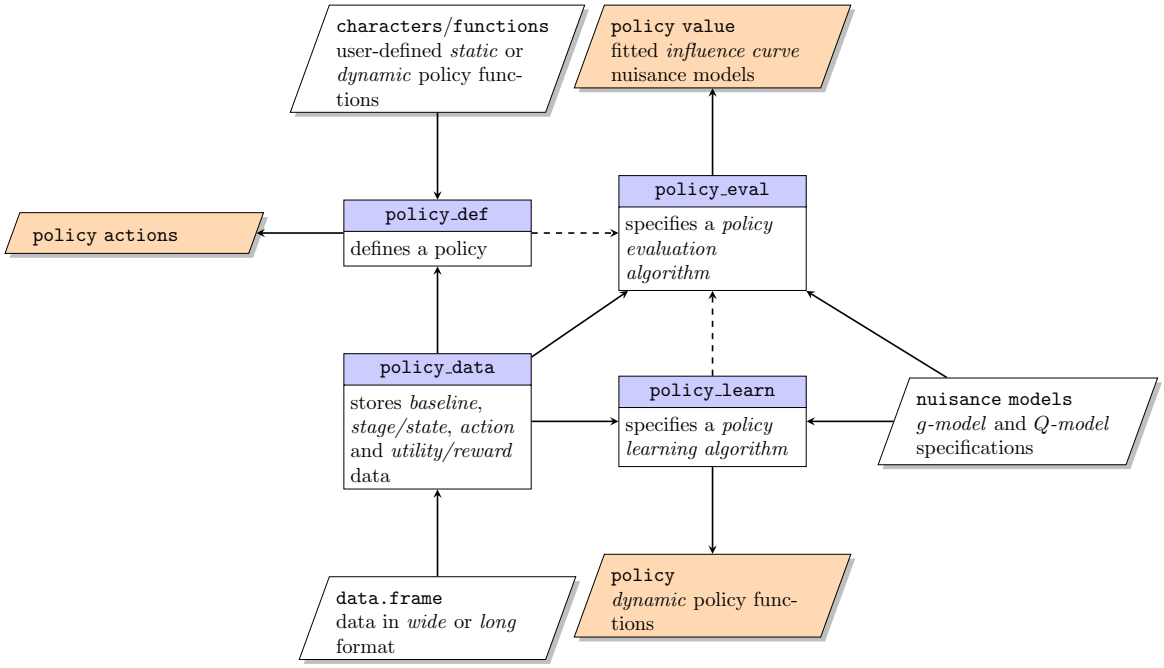


Figure 3: Overview of the four main functions of **polle** and their arguments and return values. The starting point is to define the input data in the correct format using `policy_data()`. A policy can subsequently be defined directly by the user, `policy_def`(), or estimated with one of the algorithms descripted in Section 3.3 with `policy_learn()`. The value of a policy can be estimated directly using `policy_eval()`.

object. The data input can be on long or wide format. Usually, the wide format is used for applications with a fixed number of stages and a possibly varying set of state covariates. Assume that the observed data has the sequential form

$$O = (B, X_1, U_1, A_1, X_2, W_2, U_2, A_2, U_3),$$

where B is a baseline covariate, X_1, X_2, and W_2 are state covariates, U_1, U_2, and U_3 are rewards, and A_1 and A_2 are actions. Given a `data.table`/`data.frame` denoted `data` with variable/column names `B`, `X_1`, `U_1`, `A_1`, `X_2`, `W_2`, `U_2`, `A_2` and `U_3`, we can apply `policy_data` in the following way:

```
policy_data(data,
            action = c("A_1", "A_2"),
            baseline = c("B"),
            covariates = list(X = c("X_1", "X_2"),
                              W = c(NA, "W_2")),
```

```
            utility = c("U_1", "U_2", "U_3"),
            type = "wide")
```

If only the final utility `U` is provided, we may replace `c("U_1", "U_2", "U_3")` with `c("U")`. Note that each row in `data` corresponds to a single observation.

The long format is inspired by the data format used for survival data (Therneau 2023). This format is relevant for handling a high and possibly stochastic number of stages. Assume that the observed data has the sequential form

$$O = (B, X_1, U_1, A_1, \ldots, X_{K^*}, U_{K^*}, A_{K^*}, U_{(K^*+1)}),$$

where K^* is the (stochastic) number of stages. Assume that `stage_data` is a `data.table` with variable names `id`, `stage`, `event`, `X`, `U` and `A`. `id` and `stage` denotes the observation ID and stage number. The variable `event` is an event indicator which is `0` in stage 1 through K^* and `1` in stage (K^*+1). Also assume that `baseline_data` is a `data.table` with variable names `id` and `B`. An application of `policy_data()` is now given by:

```
policy_data(stage_data,
            baseline_data = baseline_data,
            action = "A",
            baseline = c("B"),
            covariates = c("X"),
            utility = "U",
            id = "id",
            stage = "stage",
            event = "event",
            type = "long")
```

Note, an observation with K^* stages spans over (K^*+1) rows in `stage_data` and a single row in `baseline_data`.

The function `policy_def()` constructs a user-specified static or dynamic policy. The resulting policy object can be applied directly on a policy data object or as input to `policy_eval()`.

```
policy_def(policy_functions,
           full_history = FALSE,
           replicate = FALSE)
```

`policy_functions` may be a single function/character string or a list of functions/character strings defining the policy at each stage. The argument `full_history` defines the input to the policy functions. If `full_history = FALSE` the state/Markov type history (B, X_k) is passed on to the functions with variable names `B` and `X`. If `full_history = TRUE`, the full history $(B, X_1, A_1, \ldots, X_{k-1}, A_{k-1}, X_k)$ with variable names `B, X_1, A_1,..., X_(k-1), A_(k-1), X_k` is passed on to the functions. As an example, `function(X) 1*(X>0)` in combination with `full_history = FALSE` defines the policy $A_k = I\{X_k > 0\}$. Similarly, at stage $k = 2$, `function(X_1, X_2) 1*(X_1>0)*(X_2>0)` in combination with `full_history = TRUE` defines the policy $A_2 = I\{X_1 > 0, X_2 > 0\}$. The input `replicate = TRUE` will reuse the provided policy functions at each stage if possible.

`policy_learn()` specifies a policy learning algorithm which can be used directly on a policy data object or as input to `policy_eval()`. The `type` argument selects the method. Table 1 provides an overview of the method types, dependencies and limitations.

`type` argument	Method	Imports	Limitations
`"ql"`	Q-learning		
`"drql"`	Doubly Robust Q-learning. Algorithm 3, 5.		
`"ptl"`	Policy tree learning. Algorithm 2.	**policytree**	Realistic policy learning implemented for dichotomous action sets.
`"owl"`	Outcome weighted learning	**DTRlearn2**	No realistic policy learning. Fixed number of stages. Dichotomous action set. Augmentation terms are not cross-fitted.
`"earl"`	Efficient augmented and relaxation learning	**DynTxRegime**	Single stage. No cross-fitting. No realistic policy learning. Dichotomous action set.
`"rwl"`	Residual weighted learning	**DynTxRegime**	Same as `"earl"`.

Table 1: Overview of policy learning methods and their dependencies and limitations.

A cross-fitted doubly robust V-restricted Q-learning algorithm, see Algorithm 3 and 5, may be specified as follows:

```
policy_learn(type = "drql",
             control = list(qv_models = q_glm(~X)),
             full_history = FALSE,
             alpha = 0.05,
             L = 10)
```

The control argument `qv_models` is a single model or a list of models specifying the QV-models. We will subsequently describe these models in detail. Note that a QV-model is fitted for each action in the action set. The argument `full_history` specifies the history available to the QV-models similar to `full_history` in `policy_def()`. The argument `alpha` is the probability threshold for defining the set of realistic actions. The default value is `alpha = 0`. Finally, the argument `L` is the number of folds used in the cross-fitting procedure.

Similarly, a cross-fitted doubly robust sequential value search procedure based on decision trees, see Algorithm 2, may be specified as follows:

```
policy_learn(type = "ptl",
             control = control_ptl(policy_vars = c("X"),
```

```
                              depth,
                              split.step,
                              min.node.size,
                              hybrid,
                              search.depth)
            full_history = FALSE,
            alpha = 0.05,
            L = 10)
```

The function `control_ptl()` helps set the default control arguments for `type = "ptl"`. Similar functions are available for every policy learning type. The control argument `policy_vars` is a character vector or a list of character vectors further subsetting the history available to the decision tree model. The control arguments `depth`, `split.step`, `min.node.size`, and `search.depth` are directly passed on to `policytree::policy_tree()`. Each of these arguments must be an integer or an integer vector. The control argument `hybrid` is a logical value indicating whether to use `policytree::policy_tree()` or `policytree::hybrid_policy_tree()`.

The value of a user-specified policy or a policy learning algorithm can be estimated using `policy_eval()`. The evaluation can be based on inverse probability weighting or outcome regression. However, the default is to use the doubly robust value estimator given by Algorithm 4:

```
policy_eval(type = "dr",
            policy_data,
            policy,
            policy_learn,
            g_models = g_glm(~ X+B),
            g_full_history = FALSE,
            q_models = q_glm(~ A*X),
            q_full_history = FALSE,
            M = 10)
```

`g_models` and `g_full_history` specifies the modelling of the g-functions. If `g_full_history = FALSE` and a single g-model is provided, a single Markov type model across all stages is fitted. In this case a generalized linear model is fitted with a model matrix given the formula `~ X+B`. If `g_full_history = TRUE` or `g_models` is a list, a g-function is fitted for each stage. Similarly, `q_models` and `q_full_history` specifies the modelling of the Q-functions. A model is fitted at each stage. If `q_full_history = FALSE` and a single Q-model is provided, the model is reused at each stage with the same design. Alternatives to `g_glm()` and `q_glm()` are listed in Table 2. The models are created to save the design specifications, which is useful for cross-fitting. `M` is the number of folds in the cross-fitting procedure.

Call	Method	Imports	Limitations
`g_empir`	empirical (conditional) probabilities		
`g_glm/q_glm`	generalized linear model	**stats**	`g_glm`: dichotomous actions
`g_glmnet/q_glmnet`	lasso and elastic-net regularized generalized linear models	**glmnet**	`g_glmnet`: dichotomous actions
`g_rf/q_rf`	random forests	**ranger**	
`g_sl/q_sl`	super learner prediction algorithm	**SuperLearner**	`g_sl`: dichotomous actions

Table 2: Overview of available g-model and Q-model constructors.

5. Examples

In this section we go through a number of reproducible examples based on simulated data sets that illustrates different applications of the **polle** package.

In Section 5.1 we consider a single-stage problem. We demonstrate how the policy evaluation framework handles static policies to obtain estimates of causal effects. Furthermore, we evaluate the true optimal dynamic policy using highly adaptive nuisance models and use doubly robust V-restricted Q-learning to obtain an estimate of the same optimal policy. In Section 5.2 we study a problem with two fixed stages. We show how to create a policy data object from raw data on wide format and how to formulate the optimal dynamic policy over multiple stages. We use g-models and Q-models with custom data designs to evaluate the policy and showcase the use of policy trees.

The problem of having a stochastic number of stages is showcased in Section 5.3. A policy data object is created using raw data on long format. We showcase the use of partial policies and estimate the optimal partial realistic policy using doubly robust Q-learning. We show how to implement and simulate new data based on the estimated policy.

Finally, in Section 5.4 we examplify how **polle** handles problems with multiple actions in the action set.

5.1. Single-stage problem

To illustrate the usage of the **polle** package we first consider a single-stage problem. Here we consider data from a simulation where the optimal policy is known. We consider observed data from the directed acyclic graph (DAG) given in Figure 4.

The utility/reward/response is in this example defined as the conditional Gaussian distribution

$$U \mid Z, L, A \sim \mathcal{N}(Z + L + A \cdot \{\gamma Z + \alpha L + \beta\}, \sigma^2)$$

with independent state covariates/variables $Z, L \sim \text{Uniform}([0, 1])$, and treatment, A, defined by the logistic regression model

$$A \mid Z, L, B \sim \text{Bernoulli}(\text{expit}\{\kappa Z^{-2}(Z + L - 1) + \delta B\})$$

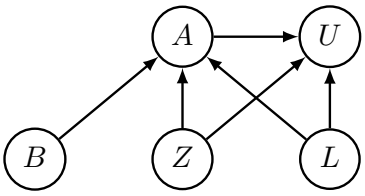


Figure 4: Single-stage problem with treatment variable A, utility U, and confounders B, Z, L.

where $B \sim \text{Bernoulli}(\pi)$ is an additional independent state covariate, and expit is the inverse logistic link function. Here we consider the choices $\pi = 0.3, \kappa = 0.1, \Delta = 0.5, \alpha = 1, \beta = -2.5, \gamma = 3, \sigma = 1$:

```
R> library("polle")
R> par0 <- c(p = .3, k = .1,  d = .5, a = 1, b = -2.5, c = 3)
R> (d <- sim_single_stage(n = 5e2, seed=1, par=par0))
```

```
           Z          L B A          U
1  1.2879704 -1.4795962 0 1 -0.9337648
2  1.6184181  1.2966436 0 1  6.7506026
3  1.2710352 -1.0431352 0 1 -0.3377580
4 -0.2157605  0.1198224 1 0  1.4993427
```

The data is first transformed using `policy_data()` with instructions on which variables defines the *action*, state *covariates* and the *utility*:

```
R> pd <- policy_data(d, action="A", covariates=list("Z", "B", "L"), utility="U")
R> pd
```

```
Policy data with n = 500 observations and maximal K = 1 stages.

     action
stage   0   1   n
    1 278 222 500

Baseline covariates:
State covariates: Z, B, L
Average utility: -0.98
```

Policy Evaluation

A single-stage *policy* is mapping from the history $H = (B, Z, L)$ onto the set of actions $\mathcal{A} = \{0, 1\}$. It is possible to evaluate both user-defined policies as well as learning a policy from the data using **polle**. Here we first illustrate how to estimate the value of a *static policy* where all individuals are given action '1' irrespective of their covariate values. Policies are defined using `policy_def()` which expects a function as input or, as here, a numeric vector specifying the static policy:

```
R> p1 <- policy_def(1, name="A=1")
R> p1

Policy with argument(s)
policy_data
```

The policy can be applied to a `policy_data` object to get the individual actions:

```
R> p1(pd)

   id stage d
1:  1     1 1
2:  2     1 1
3:  3     1 1
```

The value of the policy can then be estimated using `policy_eval()`:

```
R> (pe1 <- policy_eval(pd, policy=p1))

    Estimate Std.Err   2.5% 97.5%   P-value
A=1   -2.674  0.2116 -3.089 -2.26 1.331e-36
```

This provides an estimate of the average potential outcome, $\mathsf{E}[U^{(a=1)}]$. By default, a doubly robust estimator given by Algorithm 1 without cross-fitting is used to estimate the value. A logistic regression model with all main effects is used to model the g-function and a linear regression model with all interaction effects between the action and each of the state covariates is used to model the Q-function. We will later revisit how to estimate the value of the policy using flexible machine learning models and cross-fitting.

In the same way, we can estimate the value under the action '0':

```
R> (pe0 <- policy_eval(pd, policy=policy_def(0, name = "A=0")))

    Estimate Std.Err    2.5%  97.5% P-value
A=0 -0.02243 0.08326 -0.1856 0.1408  0.7877
```

Finally, the average treatment effect, $ATE := \mathsf{E}\{U^{(a=1)}) - U^{(a=0)}\}$, can then be estimated as:

```
R> estimate(merge(pe0, pe1), function(x) x[2]-x[1], labels="ATE")

    Estimate Std.Err   2.5%  97.5%   P-value
ATE   -2.652  0.1737 -2.992 -2.312 1.236e-52
```

The function `lava::merge.estimate()` (Holst and Budtz-Jørgensen 2013) combines the influence curve estimates for each estimate. The influence curve matrix is available via `IC()`:

```
R> IC(merge(pe0, pe1))
```

```
          A=0       A=1
1 -0.08832404 0.6568576
2  2.61912976 9.6387445
3  0.27539152 1.0710632
```

The standard errors for the transformation $f(x_1, x_2) = x_2 - x_1$ is then given by the delta method.

In this case, we know that the optimal decision boundary is defined by the hyper-plane $\gamma Z + \alpha L + \beta = 0$. Again, we use `policy_def()` to define the optimal policy:

```
R> p_opt <- policy_def(
+    function(Z, L) 1*((par0["c"]*Z + par0["a"]*L + par0["b"])>0),
+    name="optimal")
```

We estimate the value of the optimal policy using Algorithm 1. Specifically, we use `M` fold cross-fitting and super learners for the g-function and Q-function including random forests regression and generalized additive models as implemented in the **SuperLearner** package (Polley, LeDell, Kennedy, and van der Laan 2021):

```
R> set.seed(1)
R> policy_eval(
+    pd,
+    policy = p_opt,
+    g_models = g_sl(SL.library = c("SL.glm", "SL.ranger", "SL.gam")),
+    q_models = q_sl(SL.library = c("SL.glm", "SL.ranger", "SL.gam")),
+    M = 5
+  )

        Estimate Std.Err   2.5%  97.5%   P-value
optimal   0.3726  0.1109 0.1553 0.5899 0.0007793
```

Policy learning

In real applications the optimal policy is of course not known. Instead we seek to estimate/learn the optimal policy from the data. The function `policy_learn()` constructs a policy learner. Here we specify a cross-fitted doubly robust V-restricted Q-learning algorithm as given by Algorithm 3:

```
R> pl <- policy_learn(
+    type = "drql",
+    L = 5,
+    control = control_drql(qv_models = q_glm(formula = ~ Z + L))
+  )
```

The policy learner is restricted to $V = (Z, L)$ given by the `formula` argument. Remember that `L` is the number of cross-fitting folds. The algorithm can be applied directly resulting in a policy object:

```
R> set.seed(1)
R> po <- pl(
+    pd,
+    g_models = g_sl(SL.library = c("SL.glm", "SL.ranger", "SL.gam")),
+    q_models = q_sl(SL.library = c("SL.glm", "SL.ranger", "SL.gam"))
+  )
R> po

Policy object with list elements:
qv_functions, action_set, stage_action_sets, alpha, K, folds
Use 'get_policy' to get the associated policy.
```

The actions of the learned policy are available through the `get_policy()`:

```
R> get_policy(po)(pd)

   id stage d
1:  1     1 1
2:  2     1 1
3:  3     1 1
```

The value of the learned policy can also be estimated directly via `policy_eval()`:

```
R> set.seed(1)
R> pe <- policy_eval(
+    pd,
+    policy_learn = pl,
+    g_models = g_sl(SL.library = c("SL.glm", "SL.ranger", "SL.gam")),
+    q_models = q_sl(SL.library = c("SL.glm", "SL.ranger", "SL.gam")),
+    M = 5
+  )
R> pe

     Estimate Std.Err   2.5%  97.5%   P-value
drql   0.3896  0.1109 0.1723 0.6069 0.0004417
```

Note that the cross-fitting procedure is nested in this case, i.e., $M \times L$ g-functions and Q-functions are fitted. The resulting policy actions are displayed in Figure 5 along with the true optimal decision boundary.

5.2. Two-stage problem

In this example we consider a two-stage problem. An observation can be written as $O := (S_1, A_1, S_2, A_2, S_3)$, where $S_1 = (C_1, L_1, U_1)$, $S_2 = (C_2, L_2, U_2)$, and $S_3 = (L_3, U_3)$. The state covariates (cost C_k and load L_k) and action variables (A_k) are associated with the DAG in Figure 6.

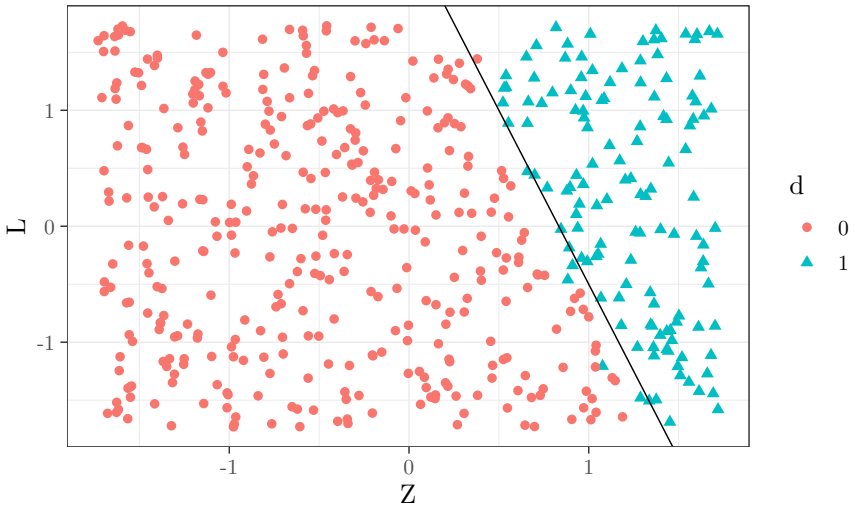


Figure 5: Fitted policy actions based on doubly robust V-restricted Q- learning. The black line shows the true optimal decision boundary.

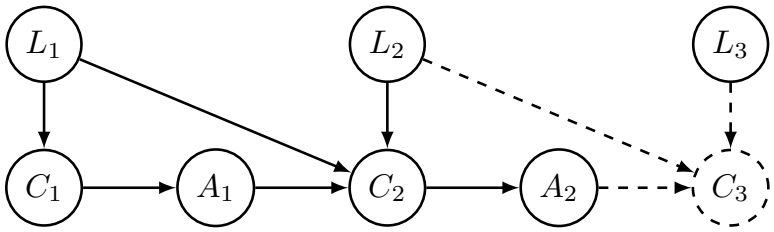


Figure 6: Two-stage problem with states $(L_1, C_1), (L_2, C_2), L_3$ with exogenous component $L_k, k = 1, 2, 3$. As the utility in this example does not depend on the last endogenous state, C_3, it can be omitted from the analysis.

Specifically, the costs, loads and actions are given by the structural model

$$\begin{aligned} L_1 &\sim \mathcal{N}(0, 1) \\ C_1 \mid L_1 &\sim \mathcal{N}(L_1, 1) \\ A_1 \mid C_1 &\sim \text{Bernoulli}(\text{expit}(\beta C_1)) \\ L_2 &\sim \mathcal{N}(0, 1) \\ C_2 \mid A_1, L_1 &\sim \mathcal{N}(\gamma L_1 + A_1, 1) \\ A_2 \mid C_2 &\sim \text{Bernoulli}(\text{expit}(\beta C_2)) \\ L_3 &\sim \mathcal{N}(0, 1) \end{aligned}$$

for parameters $\gamma, \beta \in \mathbb{R}$. The rewards are given by

$$\begin{aligned} U_1 &= L_1 \\ U_2 &= A_1 \cdot C_1 + L_2 \\ U_3 &= A_2 \cdot C_2 + L_3 \end{aligned}$$

Remember that the utility is the sum of the rewards, i.e., $U = U_1 + U_2 + U_3$. In this problem we consider the parameter choices $\gamma = 0.5, \beta = 1$:

```
R> par0 <- c(gamma = 0.5, beta = 1)
R> (d <- sim_two_stage(2e3, seed=1, par=par0))
```

```
          L_1       C_1 A_1        L_2       C_2 A_2         L_3
1:  0.9696772  1.711279   1 -0.7393434  2.424370   1 -0.83124340
2: -2.1994065 -2.643124   0  0.4828756 -2.664728   0 -0.07151015
3:  1.9480938  2.061934   0  0.4803055  2.474761   1  0.40785209
          U_1       U_2         U_3
1:  0.9696772 0.9719356  1.59312684
2: -2.1994065 0.4828756 -0.07151015
3:  1.9480938 0.4803055  2.88261357
```

The data is transformed using `policy_data()` with instructions on which variables define the *actions*, *covariates* and the *rewards* at each stage.

```
R> pd <- policy_data(d,
+                    action = c("A_1", "A_2"),
+                    covariates = list(L = c("L_1", "L_2"),
+                                      C = c("C_1", "C_2")),
+                    utility = c("U_1", "U_2", "U_3"))
R> pd
```

```
Policy data with n = 2000 observations and maximal K = 2 stages.

     action
stage    0    1    n
    1 1017  983 2000
    2  819 1181 2000

Baseline covariates:
State covariates: L, C
Average utility: 0.84
```

Policy Evaluation

The optimal policy $d_0 = (d_{0,1}, d_{0,2})$ is identified via the Q-functions. At stage 2, the Q-function is given by

$$\begin{aligned} Q_{0,2}(h_2, a_2) =& E[U \mid H_2 = h_2, A_2 = a_2] \\ =& l_1 + a_1 c_1 + l_2 + a_2 c_2. \end{aligned}$$

Thus, the optimal policy at stage 2 is

$$\begin{aligned} d_{0,2}(h_2) =& \arg \max_{a_2 \in \{0,1\}} Q_2(h_2, a_2) \\ =& I\{c_2 > 0\}. \end{aligned}$$

At stage 1, the Q-function under the optimal policy at stage 2 is given by

$$\begin{aligned} Q_{0,1}^{d_0}(h_1, a_1) =& E[Q_{0,2}(H_2, d_{0,2}(h_2)) \mid H_1 = h_1, A_a = a_1] \\ =& l_1 + a_1 c_1 + E[I\{C_2 > 0\} C_2 \mid L_1 = l_1, A_1 = a_1]. \end{aligned}$$

Let

$$\begin{aligned} \kappa(a_1, l_1) =& E[I\{C_2 > 0\} C_2 \mid L_1 = l_1, A_1 = a_1] \\ =& E[I\{C_2 > 0\} \mid L_1 = l_1, A_1 = a_1] E[C_2 \mid L_1 = l_1, A_1 = a_1, C_2 > 0] \\ =& (1 - \Phi(-\{\gamma l_1 + a_1\})) \left(\gamma l_1 + a_1 + \frac{\phi(-\{\gamma l_1 + a_1\})}{1 - \Phi(-\{\gamma l_1 + a_1\})} \right). \end{aligned}$$

The optimal policy at stage 1 can now be written as

$$\begin{aligned} d_{0,1}(h_1) =& \arg \max_{a_1 \in \{0,1\}} Q_1(h_1, a_1) \\ =& I\{(c_1 + \kappa(1, l_1) - \kappa(0, l_1)) > 0\}. \end{aligned}$$

The basis for defining the optimal policy are the histories $H_1 = (L_1, C_1)$ and $H_2 = (L_1, C_1, A_1, L_2, C_2)$ which are available via `get_history()`:

```
R> get_history(pd, stage = 1, full_history = TRUE)$H
R> get_history(pd, stage = 2, full_history = TRUE)$H

   id stage        L_1       C_1
1:  1     1  0.9696772  1.711279
2:  2     1 -2.1994065 -2.643124
3:  3     1  1.9480938  2.061934

   id stage A_1        L_1        L_2       C_1       C_2
1:  1     2   1  0.9696772 -0.7393434  1.711279  2.424370
2:  2     2   0 -2.1994065  0.4828756 -2.643124 -2.664728
3:  3     2   0  1.9480938  0.4803055  2.061934  2.474761
```

We use the `policy_def()` function to define the optimal policy:

```
R> kappa <- function(mu){
+    pnorm(q = -mu, lower.tail = FALSE) *
+      (mu + dnorm(-mu) / pnorm(-mu, lower.tail = FALSE))
+  }
R> p_opt <- policy_def(
+    list(function(C_1, L_1){
+      1*((C_1 +
+         kappa(par0[["gamma"]] * L_1 + 1) -
+         kappa(par0[["gamma"]] * L_1)
+      ) > 0)
+    },
```

```
+    function(C_2){
+      1*(C_2 > 0)
+    }),
+    full_history = TRUE,
+    name = "optimal"
+  )
R> p_opt

Policy with argument(s)
policy_data
```

The optimal policy can be applied directly on the policy data:

```
R> p_opt(pd)

   id stage d
1:  1     1 1
2:  1     2 1
3:  2     1 0
```

Doubly robust evaluation of the optimal policy requires modelling the g-functions and Q-functions. In this case, the g-function is repeated at each stage. Thus, we may combine (C_1, A_1) and (C_2, A_2) when fitting the g-function. The combined state histories and actions are available through the `get_history()` function with `full_history = FALSE`:

```
R> get_history(pd, full_history = FALSE)$H
R> get_history(pd, full_history = FALSE)$A

   id stage          L         C
1:  1     1  0.9696772  1.711279
2:  1     2 -0.7393434  2.424370
3:  2     1 -2.1994065 -2.643124

   id stage A
1:  1     1 1
2:  1     2 1
3:  2     1 0
```

Similarly, when using `policy_eval()`, we can specify the structure of the used histories:

```
R> pe_opt <- policy_eval(pd,
+                        policy = p_opt,
+                        g_models = g_glm(),
+                        g_full_history = FALSE,
+                        q_models = list(q_glm(), q_glm()),
+                        q_full_history = TRUE)
R> pe_opt
```

```
        Estimate Std.Err  2.5% 97.5%   P-value
optimal    1.311 0.06578 1.182  1.44 2.067e-88
```

On closer inspection we see that a single g-model has been fitted across all stages:

```
R> get_g_functions(pe_opt)

$all_stages
$model

Call:  NULL

Coefficients:
(Intercept)            L            C
    0.01591      0.03145      0.98013

Degrees of Freedom: 3999 Total (i.e. Null);  3997 Residual
Null Deviance:            5518
Residual Deviance: 4361         AIC: 4367
```

If `q_models` is not a list, the provided model is reused at each stage. In this case the full history is used at both stages:

```
R> get_q_functions(pe_opt)

$stage_1
$model

Call:  NULL

Coefficients:
(Intercept)           A1          L_1          C_1       A1:L_1
    0.52415      0.44348      0.17462      0.09043      0.21854
     A1:C_1
    0.92899

Degrees of Freedom: 1999 Total (i.e. Null);  1994 Residual
Null Deviance:            6029
Residual Deviance: 2772         AIC: 6343


$stage_2
$model

Call:  NULL

Coefficients:
```

```
(Intercept)           A1         A_11          L_1          L_2
  -0.014697     0.140420    -0.148007    -0.118571     0.002233
        C_1          C_2      A1:A_11       A1:L_1       A1:L_2
   0.124817    -0.031178     0.046876     0.171946     0.023246
     A1:C_1       A1:C_2
  -0.113314     0.944778

Degrees of Freedom: 1999 Total (i.e. Null);  1988 Residual
Null Deviance:            3580
Residual Deviance: 1881         AIC: 5579
```

Note that in practice only the residual value is used as input to the (residual) Q-models, i.e.,

$$\begin{aligned} Q_{0,2,\text{res}}(h_2, a_2) :=& E[U_3 \mid H_2 = h_2, A_2 = a_2] \\ =& a_2 c_2 \\ Q^{d_0}_{0,1,\text{res}}(h_1, a_1) :=& E[U_2 + Q_{0,2,\text{res}}(H_2, d_{0,2}(h_2)) \mid H_1 = h_1, A_a = a_1] \\ =& a_1 c_1 + E[I\{C_2 > 0\} C_2 \mid L_1 = l_1, A_1 = a_1]. \end{aligned}$$

The fitted values of the g-functions and Q-functions are easily extracted using `predict()`:

```
R> predict(get_g_functions(pe_opt), pd)
R> predict(get_q_functions(pe_opt), pd)
```

```
   id stage        g_0        g_1
1:  1     1 0.15139841 0.84860159
2:  1     2 0.08557919 0.91442081
3:  2     1 0.93363059 0.06636941
```

```
   id stage       Q_0       Q_1
1:  1     1  1.817896  4.063055
2:  1     2  1.800290  4.233710
3:  2     1 -2.298324 -4.790946
```

Policy Learning

A V-restricted policy can be estimated via the `policy_learn()` function. In this case we use sequential doubly robust value search based on the **policytree** package, see Algorithm 2:

```
R> pl <- policy_learn(type = "ptl",
+                     control = control_ptl(policy_vars = c("C", "L")),
+                     full_history = FALSE,
+                     L = 5)
```

The policy learner is restricted to $V = (C, L)$ given by the `policy_vars` argument. The learner can be applied directly:

```
R> po <- pl(pd,
+           g_models = g_glm(),
+           g_full_history = FALSE,
+           q_models = q_glm(),
+           q_full_history = TRUE)
R> get_policy(po)(pd)

   id stage d
1:  1     1 1
2:  1     2 1
3:  2     1 0
```

Or the value of the policy learning procedure can be estimated directly using `policy_eval()`:

```
R> set.seed(1)
R> pe <- policy_eval(pd,
+                    policy_learn = pl,
+                    g_models = g_glm(),
+                    g_full_history = FALSE,
+                    q_models = q_glm())
R> pe

    Estimate Std.Err  2.5% 97.5%   P-value
ptl    1.385  0.0809 1.226 1.544 1.068e-65
```

The associated policy objects are also saved for closer inspection, see Figure 7:

```
R> po <- get_policy_object(pe)
R> po$ptl_objects

$stage_1
policy_tree object
Tree depth:  2
Actions:  1: 0 2: 1
Variable splits:
(1) split_variable: C  split_value: -3.14122
  (2) split_variable: L  split_value: -1.67452
    (4) * action: 1
    (5) * action: 2
  (3) split_variable: C  split_value: -0.274346
    (6) * action: 1
    (7) * action: 2

$stage_2
policy_tree object
Tree depth:  2
Actions:  1: 0 2: 1
```

```
Variable splits:
(1) split_variable: C  split_value: -0.747456
  (2) split_variable: C  split_value: -0.811175
    (4) * action: 1
    (5) * action: 2
  (3) split_variable: C  split_value: 0.0237423
    (6) * action: 1
    (7) * action: 2
```

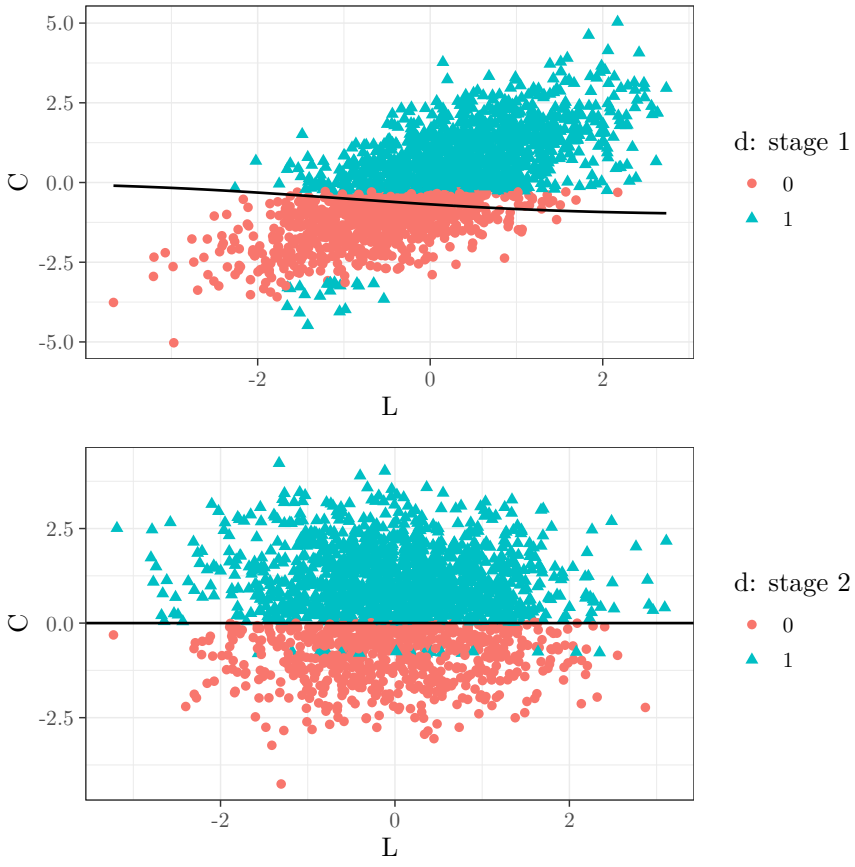


Figure 7: Fitted policy actions based on policy tree learning. The black lines show the true optimal decision boundaries.

5.3. Multi-stage problem

In this example we illustrate how **polle** handles decision processes with a stochastic number of stages. Specifically, we consider a problem with an underlying recurrent marked point process. Let $\{T_k\}_{k\geq 1}$ be a sequence of time points associated with a maintenance process for a piece of equipment. At each time point, the cost X_k and the maintenance decision $A_k \in \{0, 1\}$ are observed. If the maintenance is rejected ($A_k = 0$) the equipment is scrapped meaning that no further maintenance events occur. At baseline we observe a binary variable

B. Let W denote an unmeasured latent variable representing the quality of the equipment. For convenience, define $T_0 = 0$, $X_0 = 0$, and $A_0 = 1$. Let

$$W \sim \mathcal{N}(0, 1),$$
$$B \sim \text{Ber}(\xi),$$

and for $k \geq 1$ let

$$(T_k - T_{k-1})|X_{k-1}, A_{k-1}, W \sim \begin{cases} \text{Exp}\Big\{ \exp\left(\gamma^\top [1, X_{k-1}, W]\right) \Big\} + \psi & A_{k-1} = 1 \\ \infty & A_{k-1} = 0 \end{cases}$$
$$X_k \mid T_k, X_{k-1}, B \sim \begin{cases} \text{N}\left\{\alpha^\top [1, T_k, T_k^2, X_{k-1}, B], 1\right\} & T_k < \infty \\ 0 & T_k = \infty \end{cases}$$
$$A_k \mid X_k, T_k \sim \begin{cases} \text{Ber}\left\{\text{expit}\left(\beta^\top [1, T_k^2, X_k]\right)\right\} & T_k < \infty \\ 0 & T_k = \infty, \end{cases}$$

for parameters α, β, γ, and ψ (minimum increment). Note that if $A_k = 0$, then $T_{k+1} = \infty$, i.e., no further maintenance events occur. We consider stages within the interval $[0, \tau]$. The stochastic number of stages considered is given by

$$K^* = \max\{k : T_k \leq \tau\}.$$

We define the utility as the time in service within the interval minus the approved costs:

$$U = \sum_{k=1}^{K^*+1} U_k = \sum_{k=1}^{K^*} A_{k-1}(T_k - T_{k-1} - X_{k-1}) + A_{K^*}(\tau - T_{K^*} - X_{K^*}).$$

In this problem we consider the parameter choices $\alpha = (0, 0.5, 0.2, -0.5, 0.4)$, $\beta = (3, -0.5, -0.5)$, $\gamma = (0, -0.2, 0.3)$, $\xi = 0.3$, $\tau = 10$, $\psi = 1$:

```
R> par0 <- list(alpha =  c(0, 0.5, 0.2, -0.5, 0.4),
+               beta = c(3, -0.5, -0.5),
+               gamma = c(0, -0.2, 0.3),
+               xi = 0.3,
+               tau = 10,
+               psi = 1
+  )
R> a0 <- function(t, x, beta, ...){
+    prob <- lava::expit(beta[1] + (beta[2] * t^2) + (beta[3] * x))
+    stats::rbinom(n = 1, size = 1, prob = prob)
+  }
R> d <- sim_multi_stage(2e3, par = par0, a = a0, seed = 1)
```

The data is in long type format where the number of stages is stochastic:

```
R> d$stage_data
```

```
   id stage event        t    A          X     X_lead         U
1:  1     1     0 0.000000    1  1.3297993  0.0000000 0.0000000
2:  1     2     0 1.686561    1 -0.7926711  1.3297993 0.3567621
3:  1     3     0 3.071768    0  3.5246509 -0.7926711 2.1778778
4:  1     4     1 3.071768 <NA>         NA         NA 0.0000000
5:  2     1     0 0.000000    1  0.7635935  0.0000000 0.0000000
6:  2     2     0 1.297336    1 -0.5441694  0.7635935 0.5337427
7:  2     3     0 5.635634    0  9.0304656 -0.5441694 4.8824675
```

The data is transformed using `policy_data()` with `type = "long"`. The names of the `id`, `stage`, `event`, `action`, and `utility` (reward) variables have to be specified (in this case the default names). The event variable is 0 whenever an action occur and 1 in the terminal event. Only the utility variable is used in a terminal event.

```
R> pd <- policy_data(data = d$stage_data,
+              baseline_data = d$baseline_data,
+              type = "long",
+              id = "id",
+              stage = "stage",
+              event = "event",
+              action = "A",
+              utility = "U")
R> pd
```

```
Policy data with n = 2000 observations and maximal K = 4 stages.

     action
stage    0    1    n
    1  113 1887 2000
    2  844 1039 1883
    3  956   74 1030
    4   72    0   72

Baseline covariates: B
State covariates: t, X, X_lead
Average utility: 2.46
```

Policy Evaluation

Very few observations have more than 3 observed stages. Thus, we will only consider partial interventions on the first 3 stages. The `partial()` function is used to trim the policy data object:

```
R> pd3 <- partial(pd, K = 3)
R> pd3
```

```
Policy data with n = 2000 observations and maximal K = 3 stages.

     action
stage    0    1    n
    1  113 1887 2000
    2  844 1039 1883
    3  956   74 1030

Baseline covariates: B
State covariates: t, X, X_lead
Average utility: 2.46
```

First, we consider doubly robust evaluation of the static policy of always approving the repair ($A = 1$):

```
R> p1 <- policy_def(1, reuse = TRUE, name = "A=1")
R> pe3 <- policy_eval(pd3,
+                     policy = p1,
+                     g_models = g_glm(),
+                     q_models = q_glm(),
+                     g_full_history = FALSE,
+                     q_full_history = FALSE)
R> pe3

    Estimate Std.Err    2.5%   97.5%   P-value
A=1  -0.7208  0.1245 -0.9648 -0.4769 6.965e-09
```

When `q_full_history = FALSE` and a single Q-model is provided, the model is reused at every stage and fitted to the state/Markov type history. Here we extract the fitted Q-function for the third stage via `get_q_functions()`:

```
R> get_q_functions(pe3)[[3]]

$model

Call:  NULL

Coefficients:
(Intercept)           A1            t            X       X_lead
 -5.072e-16    1.051e+00   -5.573e-16    1.218e-16    3.303e-16
          B         A1:t         A1:X    A1:X_lead         A1:B
  1.033e-16    6.382e-01   -8.502e-02   -3.368e-01    4.581e-01

Degrees of Freedom: 1029 Total (i.e. Null);  1020 Residual
Null Deviance:            502.4
Residual Deviance: 128.4         AIC: 800.4
```

Policy Learning

The probability of a maintenance job being approved (the propensity) is high when the cost and time is low and vice versa. Figure 8 displays the fitted propensities over time t and cost X.

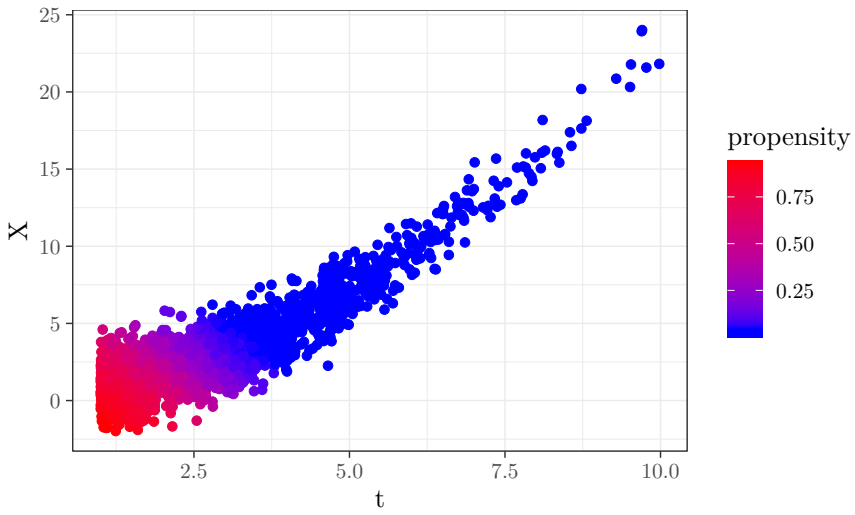


Figure 8: Fitted propensities.

In order to prevent learning poorly supported (unrealistic) policies we can use the `alpha` parameter in `policy_learn()` to set a minimum action probability of at least α:

```
R> pl <- policy_learn(type = "drql",
+                     alpha = 0.05,
+                     full_history = FALSE,
+                     control = control_drql(),
+                     L = 10)
R> set.seed(1)
R> po3 <- pl(pd3,
+            q_models = q_sl(SL.library = c("SL.mean",
+                                           "SL.glm",
+                                           "SL.glmnet",
+                                           "SL.ranger",
+                                           "SL.gam")),
+            q_full_history = FALSE,
+            g_models = g_glm(),
+            g_full_history = FALSE)
```

The resulting policy is displayed in Figure 9. We use `policy_eval()` to get the cross-fitted estimated value of the learned realistic policy:

```
R> set.seed(1)
R> future::plan("multisession")
```

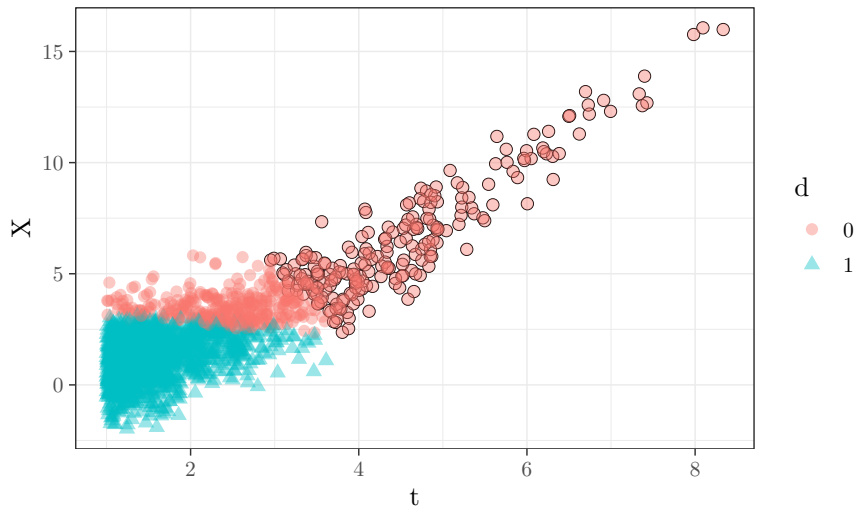


Figure 9: Optimal actions at stage 2 as dictated by a realistic QV-learning algorithm. Encircled dots indicate non-realistic actions, i.e., actions where the alternative action does not occur with probability larger than the threshold $\alpha = 0.05$.

```
R> pe3 <- policy_eval(pd3,
+                     policy_learn = pl,
+                     q_models = q_sl(SL.library =
+                                       c("SL.mean",
+                                         "SL.glm",
+                                         "SL.glmnet",
+                                         "SL.ranger",
+                                         "SL.gam")),
+                     q_full_history = FALSE,
+                     g_models = g_glm(),
+                     g_full_history = FALSE,
+                     M = 10)
R> pe3
```

```
     Estimate Std.Err  2.5% 97.5% P-value
drql    2.872 0.04867 2.777 2.968       0
```

Note that we use parallel processing via the **future** and **future.apply** packages (Bengtsson 2021). For implementation and simulation purposes, the learned policy function can also be extracted for each stage via `get_policy_functions()`. Here we extract the policy function at stage 2 and evaluate it on new custom data:

```
R> pf2 <- get_policy_functions(po3, stage = 2)
R> get_history_names(pd3)
```

```
[1] "t"      "X"      "X_lead" "B"
```

```
R> new_H <- data.frame(t = c(2, 2),
+                      X = c(1, 5),
+                      X_lead = c(0, 0),
+                      B = c(0, 0))
R> pf2(new_H)

[1] "1" "0"
```

Finally, we illustrate how to perform a simulation based on the partially fitted policy. First we extract all of the policy functions and combine it with the existing policy for all stages above the third stage:

```
R> pf <- lapply(1:3, function(x) get_policy_functions(po3, stage = x))
R> a_pf <- function(stage, t, x, beta, x_lead, b, ...){
+    if (stage <= 3){
+      H <- data.table(t = t, X = x, X_lead = x_lead, B = b)
+      out <- pf[[stage]](H)
+      return(as.numeric(out))
+    } else{
+      out <- a0(t = t, x = x, beta = beta)
+    }
+    return(out)
+  }
```

We then simulate data and estimate the value of the fitted policy:

```
R> n <- 2e4
R> d_pf <- sim_multi_stage(n, par = par0, a = a_pf, seed = 1)
R> pd_pf <- policy_data(data = d_pf$stage_data,
+                           baseline_data = d_pf$baseline_data,
+                           type = "long")
R> pd_pf

Policy data with n = 20000 observations and maximal K = 5 stages.

     action
stage     0     1     n
    1     0 20000 20000
    2  5629 14345 19974
    3 11216  2986 14202
    4  2941     5  2946
    5     5     0     5

Baseline covariates: B
State covariates: t, X, X_lead
Average utility: 2.87
```

5.4. Multi-action problems

In our final example we illustrate how **polle** handles actions sets with more than two elements. For simplicity, we consider a single stage problem. The utility is defined as the conditional Gaussian distribution

$$U \mid X, Z, A \sim \mathcal{N}\Big(X + Z + I\{A = 1\}(X^2 + Z - 0.5) + I\{A = 2\}(X - 0.5), 1\Big)$$

with independent state covariates $X, Z \sim \mathcal{N}(0, 1)$, and action variable $A \in \{0, 1, 2\}$ given as a function of a latent variable Y:

$$A = \begin{cases} 0 & \text{if} \quad Y < -1 \\ 1 & \text{if} \quad -1 \leq Y < 0.5 \\ 2 & \text{if} \quad 0.5 \leq Y, \end{cases}$$

where

$$Y|X \sim \mathcal{N}(X, 1).$$

We construct a policy data object in exactly the same way as the other single-stage problem, see Section 5.1:

```
R> d <- sim_single_stage_multi_actions(seed = 1, n = 2e3)
R> pd <- policy_data(d,
+                    action="a",
+                    covariates=c("x","z"),
+                    utility="u")
R> pd

Policy data with n = 2000 observations and maximal K = 1 stages.

     action
stage    0    1    2    n
    1  151  849 1000 2000

Baseline covariates:
State covariates: x, z
Average utility: 1.12
```

Policy Evaluation

We use `policy_def()` to define the optimal policy:

```
R> p_opt <- policy_def(
+    function(x, z){
+      i0 <- 0
+      i1 <- (x*x+z-0.5)
+      i2 <- (x-0.5)
```

```
+
+      1 * (i1>i0) * (i1 >= i2) + 2 * (i2 > i0) * (i2 > i1)
+    },
+    name = "optimal"
+  )
```

The optimal policy can now be evaluated using `policy_eval()`. The g-model input must be a multinomial classifier. The model given by `g_rf()` is a random forest classifier from the package **ranger** (Wright and Ziegler 2017).

```
R> (pe <- policy_eval(pd,
+                      policy = p_opt,
+                      g_model = g_rf(mtry = 2, num.trees = 1000),
+                      q_model = q_glm(~A*(x+z+I(x^2)))))
```

```
Loading required namespace: ranger
```

```
        Estimate Std.Err  2.5% 97.5% P-value
optimal    1.406 0.02836 1.351 1.462       0
```

The fitted g-function values are available via the functions `get_g_functions()` and `predict()`:

```
R> (g_pred <- predict(get_g_functions(pe), pd))
```

```
   id stage        g_0       g_1       g_2
1:  1     1 0.07127817 0.5229683 0.4057536
2:  2     1 0.01133056 0.5123563 0.4763131
3:  3     1 0.01321349 0.4442091 0.5425774
```

In this case (near) positivity violations is a cause for concern:

```
R> g_pred[g_0 == 0,]
```

```
    id stage g_0       g_1       g_2
1:  26     1   0 0.5835329 0.4164671
2: 310     1   0 0.2228849 0.7771151
3: 556     1   0 0.2551845 0.7448155
```

Policy Learning

Not every policy learning method can handle multiple actions (more than two), see Table 1. The available types are `"ql"`, `"drql"` and `"ptl"`. However, the type `"ptl"` can not fit realistic policies for multiple actions. Thus, we use realistic Q-learning at probability threshold $\alpha = 0.01$ with a correctly specified Q-model:

```
R> pl <- policy_learn(type="ql",
+                     alpha=0.01)
R> set.seed(1)
R> po <- pl(
+    pd,
+    g_model=g_rf(mtry = 2, num.trees = 1000),
+    q_model=q_glm(~A*(x+z+I(x^2)))
+  )
R> get_policy(po)(pd)

   id stage d
1:  1     1 1
2:  2     1 1
3:  3     1 1
```

Figure 10 display the fitted (realistic) optimal policy for each observation. The black lines indicate the true optimal decision boundaries.

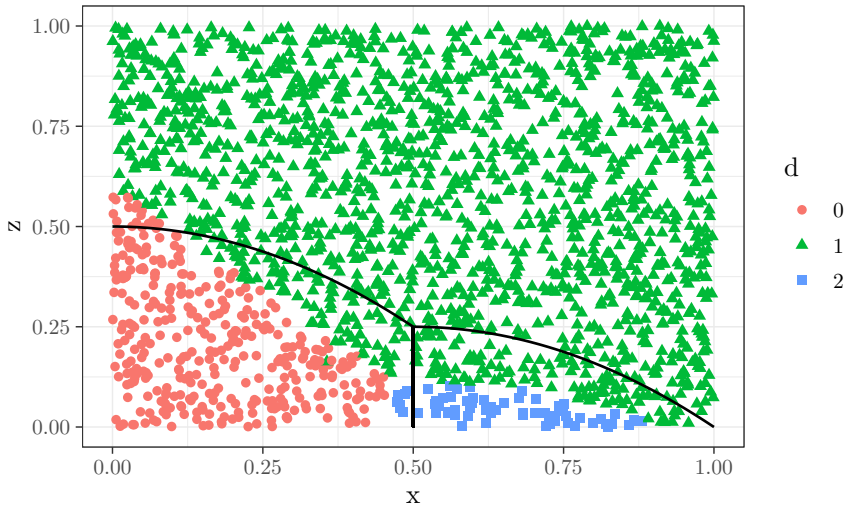


Figure 10: Fitted realistic optimal policy for each observation based on Q-learning. The black lines indicate the true optimal decision boundaries.

6. K-2 Literacy Intervention

In this section, we demonstrate the application of **polle** on a real data example. To ensure reproducibility we work with a data set publicly available in the harvard dataverse https://dataverse.harvard.edu/. Specifically, we use the kindergarten to second-grade literacy intervention data set (Kim, Asher, Burkhauser, Mesite, and Leyva 2019a) , funded by the Chan Zuckerberg initiative, which is documented and analyzed in (Kim, Asher, Burkhauser, Mesite, and Leyva 2019b). The following analysis is conducted independently of the original study, and we take full responsibility for any misinterpretations or errors.

The data set contains records of 273 students from kindergarten to second grade associated with 16 teachers. The study seeks to investigate the treatment effect of assigning two types of print texts (10 texts/books in each group) for the students to read over the summer break. All students received training before the summer break, and the investigators used a mobile app to engage and monitor each student. At stage 1 each classroom associated with a given teacher was randomly assigned to read either conceptually coherent texts (CCT) or leveled text (LT). At an intermediate time point during the summer break (stage 2), if a student had completed at least one activity on the mobile app, the student was classified as a responder and no further actions were initiated. However, all non-responders were subject to gamification of the app and the parents were randomly selected to receive text messages with reminders, information, and encouragement. The main outcome of the study was the reading measure of academic progress Rasch unit (MAP RIT) score. The score was measured before and after the summer break (referred to as spring and fall). Of the 273 students enrolled in the study, 56 students have missing MAP RIT scores. As in the original study, we conduct a complete case analysis.

We start by loading the data and conducting some basic transformations, which we include for reproducibility. Note that we define the utility as the difference in the spring and fall MAP RIT scores. A description of the variables can be found in Table 3.

```
R> library("readstata13")
R> d <- readstata13::read.dta13("k2smart_public.dta")
R> d <- transform(
+    d,
+    utility = as.numeric(fa_maprit) - as.numeric(sp_maprit),
+    cct = as.logical(cct),
+    responder = as.logical(responder),
+    text = as.logical(text),
+    maprit = as.numeric(sp_maprit),
+    teacher = as.character(t_id_public),
+    attend = as.numeric(sp_pctattend_yr),
+    trc = as.numeric(sp_trcbook_num),
+    dib = as.numeric(sp_dib_score),
+    ell = as.logical(ell),
+    iep = as.logical(iep),
+    grade = as.character(grade_final),
+    male = as.logical(male),
+    familynight = as.logical(familynight)
+  )
```

Variable	Type	Description
utility	Numeric	Difference between the spring and fall MAP RIT score
cct	Logical	If TRUE, the student receives conceptually coherent texts (CCT). If FALSE, the student receives leveled text (LT).
responder	Logical	Respondence indicator
text	Logical	If TRUE the parents received text messages.
maprit	Numeric	Spring MAP RIT score.
teacher	Character string	Teacher ID.
attend	Numeric	Student attendance percentage.
trc	Numeric	Spring text reading comprehension score.
dib	Numeric	Spring dynamic indicators of basic early literacy skills score.
ell	Logical	If TRUE, the student received English-language learner services.
iep	Logical	If TRUE, the student has an individualized education plan.
grade	Character string	Student grade: kindergarten (0), first grade (1), second grade (2).
male	Logical	If TRUE, the student is a male.
familynight	Logical	If TRUE, the family of the student attended family night.

Table 3: Variable descriptions.

As described we make a complete case analysis:

```
R> d <- subset(d,!is.na(utility))
```

Finally, we create a stage 1 and stage 2 treatment variable:

```
R> d <- transform(
+    d,
+    A_1 = ifelse(cct, "cct", "lt"),
+    A_2 = ifelse(responder, "continue", ifelse(text, "text", "notext"))
+  )
```

The `policy_data()` function is used to create a policy data object. Note that `responder` is a stage 2 state covariate:

```
R> pd <- policy_data(
+    d,
+    action = c("A_1","A_2"),
```

```
+    utility = "utility",
+    baseline=c("maprit",
+               "male",
+               "ell",
+               "iep",
+               "attend",
+               "trc",
+               "dib",
+               "grade",
+               "familynight"),
+    covariates = list(responder = c(NA, "responder"))
+  )
R> print(pd)

Policy data with n = 217 observations and maximal K = 2 stages.

     action
stage cct continue  lt notext text   n
    1 112        0 105      0    0 217
    2   0       36   0     86   95 217

Baseline covariates: maprit, male, ell, iep, attend, trc, dib, grade, familynight
State covariates: responder
Average utility: 2.7
```

Since student responders are not randomized at stage 2, only 4 static realistic policies exist:

```
R> actions <- list(
+    c("cct","text"),
+    c("cct","notext"),
+    c("lt","text"),
+    c("lt","notext")
+  )
R> static_policies <- lapply(
+    actions,
+    function(a){
+      policy_def(list(
+        function(...) a[1],
+        function(responder) ifelse(responder, "continue", a[2])
+      ),
+      name = paste(a, collapse = "_"))
+    }
+  )
R> head(static_policies[[1]](pd), 4)

   id stage    d
1:  1     1  cct
```

```
2:  1     2 text
3:  2     1  cct
4:  2     2 text
```

We begin the analysis by comparing the policy value for each of the 4 static policies. First, we consider a basic inverse probability weighting estimator:

```
R> gm <- list(g_empir(~1),
+             g_empir(~responder))
R>
R> pe_static_policies_ipw <- lapply(
+    static_policies,
+    function(p){
+      policy_eval(pd,
+                  policy = p,
+                  g_models = gm,
+                  type = "ipw")
+    }
+  )
R> do.call("merge", pe_static_policies_ipw)

           Estimate Std.Err     2.5% 97.5%  P-value
cct_text      2.920   1.159  0.64838 5.193 0.011760
----------
cct_notext    1.737   1.109 -0.43707 3.912 0.117353
----------
lt_text       3.573   1.120  1.37793 5.769 0.001422
----------
lt_notext     2.504   1.322 -0.08728 5.095 0.058233
```

Note that the reported standard errors are valid because the g-models are known in a randomized trial. The g-models specified by the function `g_empir()` computes the (conditional) empirical probabilities and match them to each student:

```
R> print(get_g_functions(pe_static_policies_ipw[[1]]))

$stage_1
$tab
     A empir_prob
1: cct   0.516129
2:  lt   0.483871

$v
character(0)


$stage_2
```

```
$tab
          A responder empir_prob
1: continue      TRUE  1.0000000
2:   notext     FALSE  0.4751381
3:     text     FALSE  0.5248619

$v
[1] "responder"


attr(,"full_history")
[1] FALSE
```

```
R> predict(get_g_functions(pe_static_policies_ipw[[1]]), pd)[c(1,2,43,44),]

   id stage    g_cct g_continue     g_lt  g_notext    g_text
1:  1     1 0.516129          0 0.483871 0.0000000 0.0000000
2:  1     2 0.000000          0 0.000000 0.4751381 0.5248619
3: 22     1 0.516129          0 0.483871 0.0000000 0.0000000
4: 22     2 0.000000          1 0.000000 0.0000000 0.0000000
```

Efficiency of the policy value estimates can be increased by using the doubly robust value scores. As we are fitting 25 sets of nuisance models we parallelize the computations via the **future.apply** package. The variable names used to specify the nuisance model are available via `get_history_names()`:

```
R> get_history_names(pd, stage = 1)

 [1] "responder_1" "maprit"      "male"        "ell"
 [5] "iep"         "attend"      "trc"         "dib"
 [9] "grade"       "familynight"

R> get_history_names(pd, stage = 2)

 [1] "A_1"         "responder_1" "responder_2" "maprit"
 [5] "male"        "ell"         "iep"         "attend"
 [9] "trc"         "dib"         "grade"       "familynight"
```

With this help we can easily specify the Q-models using the **SuperLearner** package and plug them into the `policy_eval()` function:

```
R> sl_lib <- c("SL.mean",
+              "SL.glm",
+              "SL.gam",
+              "SL.ranger",
+              "SL.nnet")
```

```
R> qm <- list(
+    q_sl(formula = ~.-responder_1, SL.library = sl_lib),
+    q_sl(formula = ~.-responder_1-responder_2, SL.library = sl_lib)
+  )

R> library("future.apply")
R> plan(list(
+    tweak("multisession", workers = 4)
+  ))
R> pe_static_policies_dr <- lapply(
+    static_policies,
+    function(p){
+      set.seed(1)
+      policy_eval(pd,
+                  policy = p,
+                  g_models = gm,
+                  q_models = qm,
+                  q_full_history = TRUE,
+                  type="dr",
+                  M = 25)
+    }
+  )
R> print(do.call("merge", pe_static_policies_dr))

R> plan("sequential")
R> print(do.call("merge", pe_static_policies_dr))

           Estimate Std.Err    2.5% 97.5%   P-value
cct_text      2.817  0.9158  1.0226 4.612 0.0020934
----------
cct_notext    2.275  1.0109  0.2936 4.256 0.0244262
----------
lt_text       3.550  1.0489  1.4938 5.606 0.0007141
----------
lt_notext     1.966  1.1535 -0.2950 4.227 0.0883399
```

So far the reported standard errors have been overly optimistic because we ignored the random teacher/classroom effect. Luckily, when working with influence curves it is easy to adjust for these types of dependencies. When estimating the variance we first sum all of the influence curve terms related to each teacher. The resulting compounded influence curve terms are then. independent and the variance is computed in the usual fashion. This approach is similar to computing clustered standard errors (Liang and Zeger 1986). The method is implemented in the `estimate()` function from the **lava** package (Holst and Budtz-Jørgensen 2013):

```
R> library("lava")
R> (est <- estimate(do.call("merge", pe_static_policies_dr), id = d$teacher))
```

```
           Estimate Std.Err    2.5% 97.5%   P-value
cct_text      2.817  1.0337  0.7915 4.843 6.418e-03
cct_notext    2.275  0.8739  0.5622 3.988 9.233e-03
lt_text       3.550  0.6538  2.2683 4.831 5.660e-08
lt_notext     1.966  1.3385 -0.6577 4.589 1.419e-01
```

As is already evident, none of the static policies are statistically different in terms of marginal value. For completeness we conduct a chi square test:

```
R> pdiff <- function(n) lava::contr(lapply(seq(n-1), \(x) seq(x, n)))
R> estimate(est, f = pdiff(4))
```

```
                          Estimate Std.Err   2.5%  97.5% P-value
[cct_text] - [cct_notext]   0.5425  0.9079 -1.237 2.3219  0.5502
[cct_text] - [lt_text]     -0.7323  1.0827 -2.854 1.3897  0.4988
[cct_text] - [lt_notext]    0.8517  1.6208 -2.325 4.0284  0.5992
[cct_notext] - [lt_text]   -1.2747  0.9051 -3.049 0.4992  0.1590
[cct_notext] - [lt_no....   0.3092  1.6607 -2.946 3.5642  0.8523
[lt_text] - [lt_notext]     1.5840  1.8128 -1.969 5.1370  0.3823

 Null Hypothesis:
  [cct_text] - [cct_notext] = 0
  [cct_text] - [lt_text] = 0
  [cct_text] - [lt_notext] = 0
  [cct_notext] - [lt_text] = 0
  [cct_notext] - [lt_notext] = 0
  [lt_text] - [lt_notext] = 0

chisq = 2.1328, df = 3, p-value = 0.5453
```

The function `conditional()` allows the user to easily compute the conditional policy value estimates based on categorical baseline covariates. Here we group by the baseline covariate `male` for the static CCT text policy:

```
R> estimate(
+    conditional(pe_static_policies_dr[[1]], pd, "male"),
+    id = d$teacher
+  )
```

```
           Estimate Std.Err   2.5% 97.5%   P-value
male:FALSE    1.874  0.6617 0.5767 3.170 4.633e-03
male:TRUE     3.943  0.6694 2.6305 5.255 3.872e-09
```

Even though none of the static policies have a marginal treatment effect we may hope to find group specific treatment effects. To investigate further, we specify a selection of doubly robust V-restricted Q-learners and estimate the cross-fitted value of the fitted policies.

We formulate simple linear QV-models using the `q_glm()` function as we do not expect to be able to find complex non-linear treatment associations in this relatively small data set.

```
R> qvm_formulas <- list(
+    qvm_1 = list(~1, ~A_1),
+    qvm_2 = list(~maprit, ~A_1+maprit),
+    qvm_3 = list(~male, ~A_1+male),
+    qvm_4 = list(~grade, ~A_1+grade),
+    qvm_5 = list(~male+maprit, ~A_1+male+maprit),
+    qvm_6 = list(~male+grade+maprit, ~A_1+grade+male+maprit)
+  )
R>
R> qvm <- lapply(qvm_formulas,
+         function(form){
+           list(q_glm(form[[1]]), q_glm(form[[2]]))
+         })
```

The Q-models are then passed to the controls of the `policy_learn()` function. Importantly, note that `alpha` is set to 0.01 in order to account for the degenerate structure of the data; A student responder always continue the treatment in stage 2.

```
R> pl_drql <- mapply(
+    qvm,
+    names(qvm),
+    FUN = function(qv, name){
+      policy_learn(type = "drql",
+                   control = control_drql(qv_models = qv),
+                   full_history = TRUE,
+                   alpha = 0.01,
+                   L = 25,
+                   cross_fit_g_models = FALSE,
+                   name = name)
+    })
```

The value of the fitted policies are cross-fitted using `policy_eval()`:

```
R> plan(list(
+    tweak("multisession", workers = 2)
+  ))
R> set.seed(1)
R> pe_drql <- lapply(
+    pl_drql,
+    function(pl){
+      set.seed(1)
+      policy_eval(pd,
+                  policy_learn = pl,
+                  g_models = gm,
+                  q_models = qm,
+                  q_full_history = TRUE,
+                  type="dr",
```

```
+                   M = 25)
+    })

R> estimate(do.call(what = "merge", unname(pe_drql)), id = d$teacher)

      Estimate Std.Err     2.5% 97.5%   P-value
qvm_1   3.0542  0.5928  1.89227 4.216 2.579e-07
qvm_2   0.9175  0.9703 -0.98436 2.819 3.444e-01
qvm_3   3.3579  1.3048  0.80053 5.915 1.007e-02
qvm_4   2.4449  1.2362  0.02206 4.868 4.795e-02
qvm_5   2.0806  0.9730  0.17365 3.988 3.248e-02
qvm_6   1.9655  1.1611 -0.31021 4.241 9.049e-02
```

None of the fitted policies show a gain in value compared to the static policy `lt_text`. However, we might still want to study a possible male treatment interaction further (`qvm_3`). We fit policy learner 3 on the complete data set and summarize the dictated actions:

```
R> set.seed(1)
R> po_drql_male <- pl_drql[["qvm_3"]](pd,
+                                     g_models = gm,
+                                     q_models = qm,
+                                     q_full_history = TRUE)
R> pa_drql_male <- get_policy(po_drql_male)(pd)
R> head(pa_drql_male, 4)

   id stage    d
1:  1     1  cct
2:  1     2 text
3:  2     1  cct
4:  2     2 text

R> pa_drql_male <- merge(pa_drql_male, get_history(pd)$H)
R> pa_drql_male[,.N, list(stage, male,d)][order(stage, male,d)]

   stage  male        d   N
1:     1 FALSE       lt 118
2:     1  TRUE      cct  99
3:     2 FALSE continue  23
4:     2 FALSE     text  95
5:     2  TRUE continue  13
6:     2  TRUE     text  86
```

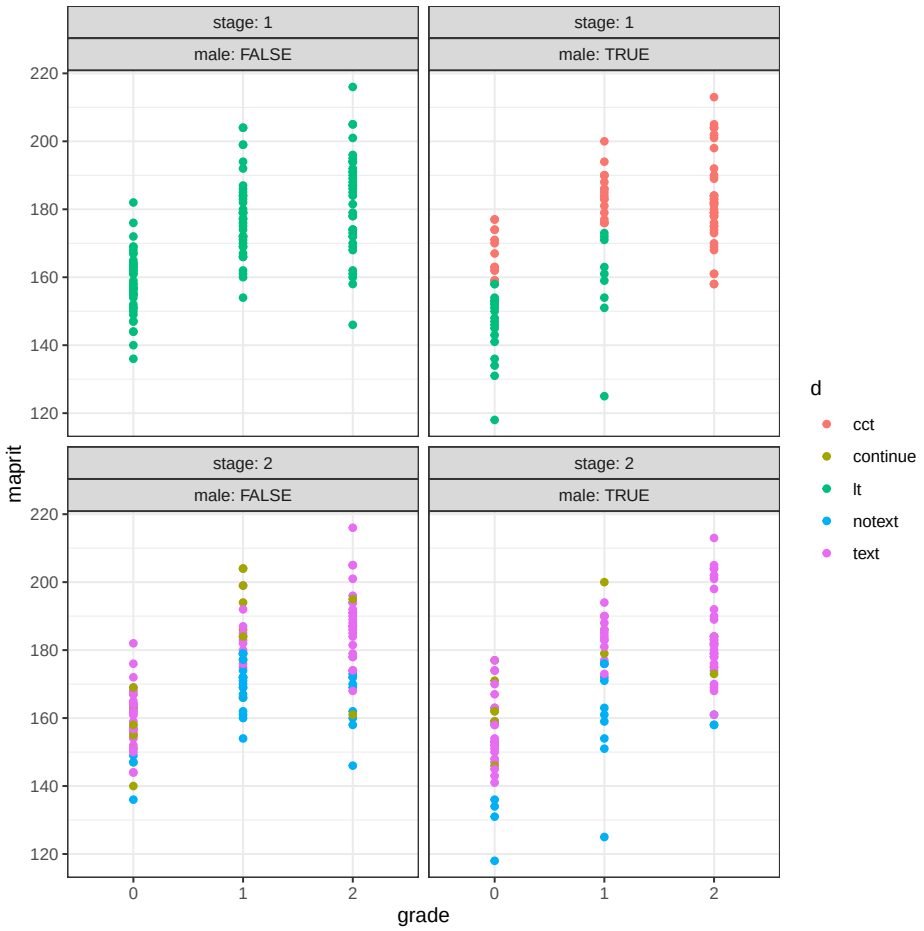
Thus, the fitted policy suggests that males receive CCT and females receive LT at stage 1 and that all non-responders get text messages at stage 2.

We end this analysis by emphasizing the importance of cross-fitting the policy learner because it is easy to overfit an optimal policy. We showcase this by fitting the most complex of the considered policy learners:

```
R> po_drql_6 <- pl_drql[["qvm_6"]](pd,
+                                  g_models = gm,
+                                  q_models = qm,
+                                  q_full_history = TRUE)
```

The dictated actions are easily plotted using the `get_history()` and `get_policy()` functions:

```
R> plot_data <- get_history(pd)$H
R> plot_data <- merge(plot_data,
+                     get_policy(po_drql_6)(pd),
+                     by = c("id", "stage"))
R> library("ggplot2")
R> ggplot(plot_data) +
+    geom_point(aes(x = grade, y = maprit, color = d)) +
+    facet_wrap(~stage+male, labeller = "label_both") +
+    theme_bw()
```



If we just plug in the resulting fitted policy we get an overly optimistic estimate of the value.

```
R> plan(list(
+    tweak("multisession", workers = 4)
```

```
+  ))
R> set.seed(1)
R> pe_plugin <- policy_eval(pd,
+                           policy = get_policy(po_drql_6),
+                           g_models = gm,
+                           q_models = qm,
+                           q_full_history = TRUE,
+                           type = "dr",
+                           M = 25,
+                           name = "qvm_6_plugin")


R> estimate(pe_plugin + pe_drql[["qvm_6"]], id = d$teacher)


             Estimate Std.Err    2.5% 97.5%   P-value
qvm_6_plugin    5.146   1.129  2.9335 7.358 5.147e-06
qvm_6           1.966   1.161 -0.3102 4.241 9.049e-02
```

7. Summary and discussion

The **polle** library is the first unifying R package for learning and evaluating policies. The package efficiently handles cross-fitting of the nuisance models and provides protection against (near) positivity violations. Also, to our knowledge, **polle** contains the first implementation of doubly robust restricted Q-learning which can serve as sensible benchmark for all other learning methods.

Of course, **polle** has its limitations. Future work to be included in the package includes the handling of missing data and (right) censored observations. The `event` variable included in the policy data object can be extended to specify missing or censored data similar to that of the `Surv` function in the **survival** package. Additional models for the censoring distribution would need to be included.

In our work we only consider the maximization of a scalar utility value. However, in some applications a multi-dimensional value vector may more naturally be of interest. In such cases the set of Pareto efficient policies can be formulated. An important example would be the task of maximizing the utility subject to variance constraints in order to learn robust policies. This is closely related to introducing a penalty term to the loss function, and will be the subject of future developments to the **polle** package.

The package is available directly from the Comprehensive R Archive Network (CRAN) (Nordland and Holst 2022). We believe the package will provide practitioners with much easier access to a broad range of policy learning methods and hope that it also may serve as a framework for benchmarking as well as implementing new methods for researchers in the policy learning field. We invite to collaboration on the future development of the package via pull requests to the github repository https://github.com/AndreasNordland/polle/.

Computational details

The results in this paper were obtained using R 4.2.1 (R Core Team 2022) with the **polle** 1.2 package. R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/.

Acknowledgments

This work was supported by a grant from InnovationsFonden Danmark (case no. 8053-00096B).

References

Athey S, Tibshirani J, Wager S (2019). "Generalized Random Forests." *The Annals of Statistics*, **47**(2), 1148–1178.

Athey S, Wager S (2021). "Policy Learning With Observational Data." *Econometrica*, **89**(1), 133–161.

Battocchi K, Dillon E, Hei M, Lewis G, Oka P, Oprescu M, Syrgkanis V (2019). "EconML: A Python Package for ML-Based Heterogeneous Treatment Effects Estimation." https://github.com/microsoft/EconML. Version 0.14.0.

Bengtsson H (2021). "A Unifying Framework for Parallel and Distributed Processing in R using Futures." *The R Journal*, **13**(2), 208–227. doi:10.32614/RJ-2021-048. URL https://doi.org/10.32614/RJ-2021-048.

Chakraborty B, Moodie E (2013). *Statistical Methods for Dynamic Treatment Regimes*. Springer-Verlag.

Chen Y, Liu Y, Zeng D, Wang Y (2020). ***DTRlearn2**: Statistical Learning Methods for Optimizing Dynamic Treatment Regimes*. R package version 1.1, URL https://CRAN.R-project.org/package=DTRlearn2.

Chernozhukov V, Chetverikov D, Demirer M, Duflo E, Hansen C, Newey W, Robins J (2018). "Double/Debiased Machine Learning For Treatment and Structural Parameters."

Ertefaie A, Almirall D, Huang L, Dziak JJ, Wagner A, Murphy S (2012). "SAS PROC QLEARN users' guide (Version 1.0.0)." URL http://methodology.psu.edu.

Goldberg Y, Kosorok MR (2012). "Q-learning With Censored Data." *Annals of statistics*, **40**(1), 529.

Hernán MA, Robins JM (2020). *Causal Inference: What If*. Boca Raton: Chapman Hall/CRC.

Hines O, Dukes O, Diaz-Ordaz K, Vansteelandt S (2022). "Demystifying Statistical Learning Based on Efficient Influence Functions." *The American Statistician*, **76**(3), 292–304.

Hirano K, Porter JR (2012). "Impossibility Results for Nondifferentiable Functionals." *Econometrica*, **80**(4), 1769–1790.

Holloway ST, Laber EB, Linn KA, Zhang B, Davidian M, Tsiatis AA (2022). ***DynTxRegime**: Methods for Estimating Optimal Dynamic Treatment Regimes*. R package version 4.11, URL https://CRAN.R-project.org/package=DynTxRegime.

Holst KK, Budtz-Jørgensen E (2013). "Linear Latent Variable Models: the lava-package." *Computational Statistics*, **28**(4), 1385–1452.

Horvitz DG, Thompson DJ (1952). "A Generalization of Sampling Without Replacement From a Finite Universe." *Journal of the American statistical Association*, **47**(260), 663–685.

Kennedy EH (2020). "Towards Optimal Doubly Robust Estimation of Heterogeneous Causal Effects." *arXiv preprint arXiv:2004.14497*.

Kim JS, Asher CA, Burkhauser M, Mesite L, Leyva D (2019a). "Replication Data for: Using a Sequential Multiple Assignment Randomized Trial (SMART) to Develop an Adaptive K2 Literacy Intervention With Personalized Print Texts and App-Based Digital Activities." doi:10.7910/DVN/AVW6KB. URL https://doi.org/10.7910/DVN/AVW6KB.

Kim JS, Asher CA, Burkhauser M, Mesite L, Leyva D (2019b). "Using a Sequential Multiple Assignment Randomized Trial (SMART) to Develop an Adaptive K–2 literacy intervention with personalized print texts and app-based digital activities." *AERA Open*, **5**(3), 2332858419872701.

Künzel SR, Sekhon JS, Bickel PJ, Yu B (2019). "Metalearners For Estimating Heterogeneous Treatment Effects Using Machine Learning." *Proceedings of the national academy of sciences*, **116**(10), 4156–4165.

Lewis G, Syrgkanis V (2020). "Double/Debiased Machine Learning for Dynamic Treatment Effects via g-estimation." *arXiv preprint arXiv:2002.07285*.

Liang KY, Zeger SL (1986). "Longitudinal Data Analysis Using Generalized Linear Models." *Biometrika*, **73**(1), 13–22.

Luedtke A, Chambaz A (2020). "Performance Guarantees for Policy Learning." In *Annales de l'IHP Probabilites et statistiques*, volume 56, p. 2162. NIH Public Access.

Luedtke AR, Van Der Laan MJ (2016). "Statistical Inference For the Mean Outcome Under a Possibly Non-unique Optimal Treatment Strategy." *Annals of statistics*, **44**(2), 713.

Luedtke AR, van der Laan MJ (2016). "Super-learning of an Optimal Dynamic Treatment Rule." *The international journal of biostatistics*, **12**(1), 305–332.

Nordland A, Holst KK (2022). ***polle**: Policy Learning*. R package version 1.0, URL https://CRAN.R-project.org/package=polle.

Petersen ML, Porter KE, Gruber S, Wang Y, Van Der Laan MJ (2012). "Diagnosing and responding to violations in the positivity assumption." *Statistical methods in medical research*, **21**(1), 31–54.

Polley E, LeDell E, Kennedy C, van der Laan M (2021). ***SuperLearner**: Super Learner Prediction*. R package version 2.0-28, URL https://CRAN.R-project.org/package=SuperLearner.

R Core Team (2022). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Robins J (1986). "A new approach to causal inference in mortality studies with a sustained exposure periodapplication to control of the healthy worker survivor effect." *Mathematical modelling*, **7**(9-12), 1393–1512.

Robins J, Rotnitzky AG (2014). "Discussion of Dynamic treatment regimes: Technical challenges and applications."

Rubin DB (1974). "Estimating Causal Effects of Treatments in Randomized and Nonrandomized Studies." *Journal of educational Psychology*, **66**(5), 688.

Semenova V, Chernozhukov V (2021). "Debiased Machine Learning of Conditional Average Treatment Effects and Other Causal Functions." *The Econometrics Journal*, **24**(2), 264–289.

StataCorp L (2021). "StataCorp. Stata Statistical Software: Release 17." URL https://www.stata.com/.

Sverdrup E, Kanodia A, Zhou Z, Athey S, Wager S (2020). "policytree: Policy Learning via Doubly Robust Empirical Welfare Maximization Over Trees." *Journal of Open Source Software*, **5**(50), 2232.

Sverdrup E, Kanodia A, Zhou Z, Athey S, Wager S (2022). ***policytree**: Policy Learning via Doubly Robust Empirical Welfare Maximization over Trees*. R package version 1.2.0, URL https://CRAN.R-project.org/package=policytree.

Therneau TM (2023). *A Package for Survival Analysis in R*. R package version 3.5-3, URL https://CRAN.R-project.org/package=survival.

Tsiatis AA, Davidian M, Holloway ST, Laber EB (2019). *Dynamic treatment regimes: Statistical methods for precision medicine*. Chapman and Hall/CRC.

Van der Laan MJ (2006). "Statistical Inference For Variable Importance." *The International Journal of Biostatistics*, **2**(1).

van der Laan MJ, Luedtke AR (2014). "Targeted Learning of an Optimal Dynamic Treatment, and Statistical Inference For Its Mean Outcome."

Van der Laan MJ, Robins JM (2003). *Unified Methods for Censored Longitudinal Data and Causality*, volume 5. Springer-Verlag.

Wright MN, Ziegler A (2017). "**ranger**: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R." *Journal of Statistical Software*, **77**(1), 1–17. doi:10.18637/jss.v077.i01.

Zhang B, Tsiatis AA, Davidian M, Zhang M, Laber E (2012). "Estimating optimal treatment regimes from a classification perspective." *Stat*, **1**(1), 103–114.

Zhang C, Chen J, Fu H, He X, Zhao YQ, Liu Y (2020). "Multicategory Outcome Weighted Margin-based Learning For Estimating Individualized Treatment Rules." *Statistica sinica*, **30**, 1857.

Zhao Y, Zeng D, Rush AJ, Kosorok MR (2012). "Estimating individualized treatment rules using outcome weighted learning." *Journal of the American Statistical Association*, **107**(499), 1106–1118.

Zhou Z, Athey S, Wager S (2018). "Offline Multi-action Policy Learning: Generalization and Optimization. arXiv preprint arXiv." *arXiv preprint arXiv:1810.04778*.

A. Binary V-optimal policy

Consider the two-stage case, $O = (S_1, A_1, S_2, A_2, U)$, where A_1 and A_2 are binary. Let V_1 be a function of $H_1 = S_1$ and let (A_1, V_2) be a function of H_2. The V-optimal policy is defined as

$$d_0 = \arg\max_{d \in \mathcal{D}} E[U^d].$$

The following theorem is a corrected version of Theorem 1 in van der Laan and Luedtke (2014).

Theorem A.1:
If V_1 is a function of V_2, then the V-optimal policy d_0 is given by

$$\begin{aligned}
B_{0,2}(a_1, v_2) &= \mathsf{E}[U^{a_1,a_2=1} | V_2^{a_1} = v_2] - \mathsf{E}[U^{a_1,a_2=0} | V_2^{a_1} = v_2] \\
d_{0,2}(a_1, v_2) &= I\left\{B_{0,2}(a_1, v_2) > 0\right\} \\
B_{0,1}(v_1) &= \mathsf{E}[U^{a_1=1,d_{0,2}} | V_1 = v_1] - \mathsf{E}[U^{a_1=0,d_{0,2}} | V_1 = v_1] \\
d_{0,1}(v_1) &= I\left\{B_{0,1}(v_1) > 0\right\}.
\end{aligned}$$

The above statement os also true if for all a_1 and a_2

$$\mathsf{E}[U^{a_1,a_2} | V_1, V_2^{a_1}] = \mathsf{E}[U^{a_1,a_2} | V_2^{a_1}]. \tag{15}$$

Proof. Let $V^a = (V_1, V_2^a)$. For any policy d

$$\begin{aligned}
\mathsf{E}[U^d] &= \mathsf{E}\left[\sum_{a_1,a_2} U^{a_1,a_2} I\{d_2(a_1, V_2^{a_1}) = a_2\} I\{d_1(V_1) = a_1\}\right] \\
&= \sum_{a_1} \mathsf{E}\left[\left\{\sum_{a_2} E\left(U^{a_1,a_2} | V_2^{a_1}\right) I\{d_2(a_1, V_2^{a_1}) = a_2\}\right\} I\{d_1(V_1) = a_1\}\right],
\end{aligned}$$

where it is used that V_1 is a function of $V_2^{a_1}$ or that (15) holds. For any a_1 the inner sum is maximzed in d_2 by $d_{0,2}$, i.e., $\mathsf{E}[U^d] \leq E[U^{d_1,d_{0,2}}]$. Now,

$$\mathsf{E}[U^{d_1,d_{0,2}}] = \mathsf{E}\left[\sum_{a_1} \mathsf{E}[U^{a_1,d_{0,2}} | V_1] I\{d_1(V_1) = a_1\}\right],$$

which is maximized for $d_1 = d_{0,1}$, i.e., $\mathsf{E}[U^d] \leq \mathsf{E}[U^{d_1,d_{0,2}}] \leq \mathsf{E}[U^{d_{0,1},d_{0,2}}]$.

□

Note that

$$\mathsf{E}[U^{a_1,a_2=1} | V_2^{a_1} = v_2] = \mathsf{E}[U^{a_2=1} | V_2 = v_2, A_1 = a_1].$$

B. Blip loss function

We continue to consider the two-stage case from Appendix A. Assuming positivity, define the inverse probability weighted blip score as

$$D_2(g)(O) = \frac{2A_2 - 1}{g_2(H_2, A_2)} U.$$

We see that

$$\begin{aligned}
&\mathsf{E}\left[D_2(g_0)(O)|A_1, V_2\right] \\
=&\mathsf{E}\left[\frac{I\{A_2 = 1\}}{g_{0,2}(H_2, A_2)} U|A_1, V_2\right] - \mathsf{E}\left[\frac{I\{A_2 = 0\}}{g_{0,2}(H_2, A_2)} U|A_1, V_2\right] \\
=&\mathsf{E}[U^{A_2=1}|A_1, V_2] - \mathsf{E}[U^{A_2=0}|A_1, V_2] \\
=&B_{0,2}(A_1, V_2),
\end{aligned}$$

where the blip at stage 2, $B_{0,2}$, is defined in Appendix A. The V-restricted optimal policy at stage 2 is uniquely defined in terms of the blip by Theorem A.1. Similarly, define the doubly robust blip score as

$$\begin{aligned}
D_2(g, Q)(O) =& \frac{2A_2 - 1}{g_2(H_2, A_2)} \left(U - Q_2(H_2, A_2)\right) \\
&+ Q_2(H_2, 1) - Q_2(H_2, 0),
\end{aligned}$$

where $Q_{0,2}(h_2, a_2) = \mathsf{E}[U|H_2 = h_2, A_2 = a_2]$. The score is doubly robust in the sense that if $g = g_0$, then

$$\begin{aligned}
&E\left[D_2(g_0, Q)(O)|A_1, V_2\right] \\
&\quad = B_{0,2}(A_1, V_2) \\
&\quad + E\left[\frac{I\{A_2 = 1\}}{g_{0,2}(H_2, A_2)} Q_2(H_2, A_2)|A_1, V_2\right] - E\left[\frac{I\{A_2 = 0\}}{g_{0,2}(H_2, A_2)} Q_2(H_2, A_2)|A_1, V_2\right] \\
&\quad + E[Q_2(H_2, 1)|A_1, V_2] - E[Q_2(H_2, 0)|A_1, V_2] \\
&\quad = B_{0,2}(A_1, V_2) \\
&\quad + E\left[\frac{I\{A_2 = 1\}}{g_{0,2}(H_2, 1)} Q_2(H_2, 1)|A_1, V_2\right] - E\left[\frac{I\{A_2 = 0\}}{g_{0,2}(H_2, A_2)} Q_2(H_2, A_2)|A_1, V_2\right] \\
&\quad + E[Q_2(H_2, 1)|A_1, V_2] - E[Q_2(H_2, 0)|A_1, V_2] \\
&\quad = B_{0,2}(A_1, V_2),
\end{aligned}$$

and if $Q = Q_0$, then

$$\begin{aligned}
&E\left[D_2(g, Q_0)(O)|A_1, V_2\right] \\
&\quad = E\left[\frac{2A_2 - 1}{g_2(H_2, A_2)} \left(Q_{0,2}(H_2, A_2) - Q_{0,2}(H_2, A_2)\right)|A_1, V_2\right] \\
&\quad + B_{0,2}(A_1, V_2) \\
&\quad = B_{0,2}(A_1, V_2).
\end{aligned}$$

For a measurable function B_2 of (A_1, V_2) define the inverse probability weighted blip loss function as

$$L_2(B_2)(g)(O) = \left(D_2(g)(O) - B_2(A_1, V_2)\right)^2.$$

The expectation of the loss is given by

$$\begin{aligned}
\mathsf{E}[L_2(B_2)(g_0)(O)] =& \mathsf{E}\left[\left(D_2(g_0)(O) - B_2(A_1, V_2)\right)^2\right] \\
=& \mathsf{E}\left[D_2(g_0)(O)^2\right] \\
&+ \mathsf{E}\left[B_2(A_1, V_2)^2\right] \\
&- 2\mathsf{E}\left[D_2(g_0)(O) B_2(A_1, V_2)\right] \\
=& \mathsf{E}\left[D_2(g_0)(O)^2\right] \\
&+ \mathsf{E}\left[B_2(A_1, V_2)^2\right] \\
&- 2\mathsf{E}\left[B_{0,2}(A_1, V_2) B_2(A_1, V_2)\right] \\
=& \mathsf{E}\left[\left(B_2(A_1, V_2) - B_{0,2}(A_1, V_2)\right)^2\right] \\
&+ \mathsf{E}\left[D_2(g_0)(O)^2\right] \\
&- \mathsf{E}\left[B_{0,2}(A_1, V_2)^2\right].
\end{aligned}$$

The last two term are constant in B_2. Thus $\mathsf{E}[L_2(B_2)(g_0)(O)]$ is minimized in B_2 when $B_2 = B_{0,2}$ almost surely, i.e., $L_2(g_0)$ and $L_2(Q, g)$, where either $g = g_0$ or $Q = Q_0$, are valid loss function for $B_{0,2}$.

At the first stage define

$$D_1(g)(O) = \frac{2A_1 - 1}{g_1(H_1, A_1)} U.$$

For a given policy at the second stage, d_2, we see that

$$\begin{aligned}
& E\left[D_1(g_0)(O^{d_2}) | V_1 = v_1\right] \\
=& E\left[\frac{2A_1 - 1}{g_{0,2}(H_1, A_1)} U^{d_2} | V_1 = v_1\right] \\
=& E\left[U^{A_1=1, d_2} - U^{A_1=0, d_2} | V_1 = v_1\right] \\
=& B_{0,1}(v_1).
\end{aligned}$$

Thus, a valid loss function in the observed data is given by

$$L_1(B_1)(g_0, d_2)(O) = \frac{I\{A_2 = d_2(A_1, V_2)\}}{g_{0,2}(H_2, A_2)} \left(D_1(g_0)(O) - B_1(V_1)\right)^2,$$

since

$$E\left[L_1(B_1)(g_0, d_2)(O)\right] = E\left[\left(D_1(g_0)(O^{d_2}) - B_1(V_1)\right)^2\right].$$

Alternatively, for stage one define

$$D_1(g, d_2)(O) = \frac{I\{A_2 = d_2(A_1, V_2)\}}{g_2(H_2, A_2)} \frac{2A_1 - 1}{g_1(H_1, A_1)} U$$

Then

$$\begin{aligned} & E\left[D_1(g_0, d_{0,2})(O) | V_1\right] \\ =& B_{0,1}(V_1), \end{aligned}$$

and a valid loss function for $d_{0,1}$ is given by

$$\left(D_1(g_0, d_{0,2})(O) - B_1(V_1)\right)^2.$$

As in the second stage it is possible to construct a doubly robust loss function for the first stage. Define

$$\begin{aligned} D_1(g, Q^{d_2}, d_2)(O) = & Q_1^{d_2}(H_1, 1) - Q_1^{d_2}(H_1, 0) \\ & + \frac{I\{A_2 = d_2(A_1, V_2)\}}{g_2(H_2, A_2)} \frac{2A_1 - 1}{g_1(H_1, A_1)} \{U - Q_2(H_2, A_2)\} \\ & + \frac{2A_1 - 1}{g_1(H_1, A_1)} \left\{Q_2(H_2, d_2(V_2, A_1)) - Q_1^{d_2}(H_1, A_1)\right\}. \end{aligned}$$

Note that

$$\begin{aligned} Q_{0,1}^{d_2}(H_1, a_1) =& E\left(Q_{0,2}(H_2, d_2(V_2, A_1)) | H_1, A_1 = a_1\right) \\ =& E\left(E[\, U | H_2, A_2 = d_2(V_2, A_1)\,] | H_1, A_1 = a_1\right) \\ =& E\left(E[\, U^{d_2} | H_2\,] | H_1, A_1 = a_1\right) \\ =& E\left(U^{A_1 = a_1, d_2} | H_1\right). \end{aligned}$$

Thus, if $Q^{d_2} = Q^{d_0,2}$, then

$$E\left[D_1(g, Q^{d_2}, d_2)(O) | V_1\right] = B_{0,1}(V_1),$$

and if $g = g_0$, then

$$\begin{aligned}
E\left[D_1(g, Q^d, d_2)(O)|V_1\right] = & E\left[Q_1^{d_2}(H_1, 1) - Q_1^{d_2}(H_1, 0)|V_1\right] \\
& + B_{0,1}(V_1) \\
& - E\left[\frac{I\{A_2 = d_2(A_1, V_2)\}}{g_2(H_2, A_2)} \frac{2A_1 - 1}{g_1(H_1, A_1)} Q_2(H_2, A_1)\middle| V_1\right] \\
& + E\left[\frac{2A_1 - 1}{g_1(H_1, A_1)} Q_2(H_2, d_2(V_2, A_1))\middle| V_1\right] \\
& - E\left[\frac{2A_1 - 1}{g_1(H_1, A_1)} Q_1^{d_2}(H_1, A_1)\middle| V_1\right] \\
= & E\left[Q_1^{d_2}(H_1, 1) - Q_1^{d_2}(H_1, 0)|V_1\right] \\
& + B_{0,1}(V_1) \\
& - E\left[\frac{2A_1 - 1}{g_1(H_1, A_1)} Q_2(H_2, d_2(V_2, A_1))\middle| V_1\right] \\
& + E\left[\frac{2A_1 - 1}{g_1(H_1, A_1)} Q_2(H_2, d_2(V_2, A_1))\middle| V_1\right] \\
& - E\left[Q_1^{d_2}(H_1, 1) - Q_1^{d_2}(H_1, 0)|V_1\right] \\
= & B_{0,1}(V_1).
\end{aligned}$$

C. Weighted classification loss function

We continue the setup from Appendix B. At the second stage define the loss function

$$\breve{L}_2(d_2)(g_0)(O) = |D_2(g_0)(O)| I\Big\{d_2(A_1, V_2) \neq I\{D_2(g_0)(O) > 0\}\Big\}$$

This is a valid loss function since $\tilde{L}_2(d_2)(g_0)(O)$ is a valid loss function and

$$\begin{aligned}
-\tilde{L}_2(d_2)(g)(O) = & \frac{I\{A_2 = d_2(A_1, V_2)\}}{g_2(H_2, A_2)} U \\
= & d_2(A_1, V_2) D_2(g)(O) + \frac{I\{A_2 = 0\}}{g_2(H_2, A_2)} U \\
= & I\{D_2(g)(O) > 0\}|D_2(g)(O)| + \frac{I\{A_2 = 0\}}{g_2(H_2, A_2)} U \\
& - |D_2(g)(O)| I\Big\{d_2(A_1, V_2) \neq I\{D_2(g_0)(O) > 0\}\Big\}
\end{aligned}$$

The last equality holds because for $d \in \{0, 1\}$ and $D \in \mathbb{R}$

$$\begin{aligned}
dD = & d|D|I\{D > 0\} - d|D|I\{D \leq 0\} \\
= & |D|I\{D > 0\} - |D|\Big((1 - d)I\{D > 0\} + dI\{D \leq 0\}\Big) \\
= & |D|I\{D > 0\} - |D|I\Big\{d \neq I\{D > 0\}\Big\}.
\end{aligned}$$

Similarly, at stage one, a valid loss function for $d_{0,1}$ is given by

$$\hat{L}_1(d_1)(g_0, d_{0,2})(O) = |D_1(g_0, d_{0,2})(O)| I\Big\{d_1(V_1) \neq I\{D_1(g_0, d_{0,2})(O) > 0\}\Big\},$$

since

$$\begin{aligned}
-\tilde{L}_1(d_1)(g_0, d_{0,2})(O) =& \frac{I\{A_1 = d_1(V_1)\}}{g_{0,1}(H_1, A_1)} \frac{I\{A_2 = d_{0,2}(A_1, V_2)\}}{g_{0,2}(H_2, A_2)} U \\
=& d_1(V_1) D_1(g_0, d_{0,2})(O) + \frac{I\{A_1 = 0\}}{g_{0,1}(H_1, A_1)} \frac{I\{A_2 = d_{0,2}(A_1, V_2)\}}{g_{0,2}(H_2, A_2)} U \\
=& -|D_1(g_0, d_{0,2})(O)| I\Big\{d_1(V_1) \neq I\{D_1(g_0, d_{0,2})(O) > 0\}\Big\} \\
&+ \frac{I\{A_1 = 0\}}{g_{0,1}(H_1, A_1)} \frac{I\{A_2 = d_{0,2}(A_1, V_2)\}}{g_{0,2}(H_2, A_2)} U \\
&+ I\{D_1(g_0, d_{0,2})(O) > 0\} |D_1(g_0, d_{0,2})(O)|.
\end{aligned}$$

Affiliation:

Andreas Nordland
Section of Biostatistics
University of Copenhagen
Øster Farigmagsgade 5
1014 Copenhagen
Denmark
E-mail: anno@sund.ku.dk

Klaus Holst
Global Data Analytics
A.P. Moeller-Maersk
Esplanaden 50
1098 Copenhagen
Denmark
E-mail: klaus.holst@maersk.com

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd

Paper B

Realistic policy learning with application to asset maintenance

Authors:
Andreas Nordland & Klaus K. Holst

Publication details:
Not submitted

Journal of the Royal Statistical Society Series C: Applied Statistics, 2023, pp. 1–17

doi:
Original Article

ORIGINAL ARTICLE

Realistic policy learning with application to asset maintenance

Andreas Nordland[1,2,*] and Klaus K. Holst[2]

[1]Section of Biostatistics,University of Copenhagen,Øster Farimagsgade 5 opg B.,1014 Copenhagen,Denmark and
[1]A.P. Møller-Mærsk,Esplanaden 50,1098 Copenhagen,Denmark
[*]Corresponding author. anno@sund.ku.dk

Abstract

This paper introduces a novel application of assumption lean statistical policy learning for physical asset management. Our focus is on optimizing corrective maintenance policies and addressing the limited variation in the current decision process. To handle practical positivity violations, we restrict policies to dynamic realistic action sets. Through a thorough simulation study, we show the advantages of imposing positivity restrictions, especially for doubly robust policy learners and policy evaluation. Based on these results, we strongly suggest adopting positivity restrictions as a standard practice. We illustrate the effectiveness of our methods with a real-world case study on refrigerated container maintenance.

Key words: policy learning; dynamic treatment regimes; near positivity violations; causal inference; Q-learning; double machine learning; physical asset management

1. Introduction

In asset-heavy industries such as transportation and logistics, maintenance and repair policies play a crucial role in equipment management. Even minor enhancements to these policies can lead to substantial cost savings, considering the scale of operations involved. This raises an important question: Can a given set of historical data, collected over extended periods be utilized to improve existing policies?

The pursuit of learning optimal policies from data has garnered considerable attention across various fields. The reinforcement learning literature has made notable progress in on-policy or online data settings, resulting in numerous practical applications with well-known systems.

Off-policy reinforcement learning methods, such as importance sampling or Q-learning, have also enabled learning from historical data. However, all of these methods rely on a key Markov assumption that needs to be justified in each case. In a maintenance application, we can justify the Markov assumption if we can accurately measure the equipment's condition at the time of an action. However, in cases where the data detail level is insufficient to support a Markov decision problem approach, what is the minimal set of structural assumptions we need to consistently learn from historical data?

An assumption-lean approach to policy learning has seen significant advancements within the statistical and econometric literature, finding applications in health sciences, social sciences, and economics. The causal framework and the estimation approach in this paradigm differ substantially from those in Markov decision problem solutions. The purpose of this work is to bring some of these new developments to the field of maintenance policy learning.

The contribution of this paper is two-fold. Firstly, we introduce completely new methodology to the field of physical asset management by showcasing a novel application of statistical policy learning from historical data that relies on a minimal set of structural assumptions. Our focus lies in developing robust maintenance and repair policies, while assessing the associated risk of implementing these policies in terms of the expected value gain. Our methods are designed to address positivity violations arising from limited variation in the historical decision-making process due to existing guidelines. We apply our

© The Author 2023. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

methods to a real-case where we seek to estimate an optimal equipment maintenance and repair policy for refrigerated containers (commonly known as reefers) owned by the shipping company Maersk.

Secondly, the statistical policy learning literature has not adequately addressed the significant challenges associated with (near) positivity that arise in our application. We believe that these challenges warrant greater attention. To the best of our knowledge, our simulation study is the first of its kind to examine the significance of realistic recursive policy learning in cases where practical positivity violations occur across multiple stages. We believe that insights from this study may provide important guidelines for practitioners in the field of statistical policy learning.

The remainder of this paper is organized as follows. Section 2 provides the context for our work and a literature review on asset maintenance and statistical policy learning, with special emphasis on the practical challenges that we encounter. In Section 3, we introduce the fundamental concepts of a general asset maintenance optimization problem and establish the methodology used throughout the paper. Section 4 presents a comprehensive simulation study that investigates various policy learning strategies, replicating a real-world case of estimating an optimal realistic maintenance policy for reefers. In Section 5, we present the results of the actual case application. Finally, Section 6 concludes the paper by highlighting potential avenues for future research.

2. Context and Literature Review

The field of physical asset maintenance optimization encompasses several aspects, ranging from short-term availability to life-cycle cost optimization. For a comprehensive understanding of the terminology and methodologies used in this field, we refer to the works [Farinha, 2018] and [De Jonge and Scarf, 2020]. In our setup and application, we focus on corrective maintenance for a repairable single-unit system. The maintenance process is considered corrective as we do not have control over the timing of work orders associated with the unit.

A common approach in maintenance optimization is to formulate the problem as a (partially observable) Markov decision process and then apply reinforcement techniques [Puterman, 2014, Sutton and Barto, 2018, Uehara and Sun, 2023]. Recent applications of this approach can be found in [Liu et al., 2020, Barde et al., 2019, Srinivasan and Parlikad, 2014, Zhang and Si, 2020, Andriotis and Papakonstantinou, 2021]. However, all of these approaches rely on a known (simulation) system and strong structural assumptions on the states and how repairs affects these states.

Within the topic of corrective maintenance, to the best of our knowledge, no other work has considered a general causal framework which explicitly accounts for the use of historical data as done in statistical policy learning. Therefore, our application can be viewed as a novel contribution to the field of asset maintenance. Specific to our application of maintenance policies for shipping containers there is very limited research available. We refer to a technical paper by [Hoffmann et al., 2020], which describes a decision model for the maintenance of shipping containers. Considering that container shipping is the backbone of global trade we hope that this work can inspire more research in a field that can have a large impact.

For comprehensive reviews on statistical policy learning methodology and dynamic treatment regimes, we refer to [Chakraborty and Moodie, 2013, Kosorok and Laber, 2019, Tsiatis, 2019]. In this work, our main focus is on doubly robust loss-based learning of the blip functions, following the formulation by [Luedtke and van der Laan, 2016]. The concept of blips was originally introduced in the works of [Murphy, 2003] and [Robins, 2004]. Our approach falls within the broader scope of recursive policy learning, which includes other methods such as Q-learning, A-learning [Schulte et al., 2014], and outcome weighted learning [Zhao et al., 2012]. When-to-treat or when-to-sell policies [Nie et al., 2021] is a different line of work which is highly relevant for maintenance policy optimization.

A lot of recent work studies performance guarantees for policy learning in the single-stage case (heterogeneous treatment effect) [Semenova and Chernozhukov, 2021, Kennedy, 2020, Athey and Wager, 2020]. The latter of the listed works also provide an excellent overview of related research within the econometrics literature.

The positivity assumption or the overlap is a crucial assumption for ensuring the causal validity of policy learning methods. It ensures that observed actions are sufficiently randomized across the state space. Near or practical positivity violations, which have been extensively studied in the causal statistical literature [Bembom and van der Laan, 2007, Petersen et al., 2012, Cole and Hernán, 2008, Moore et al., 2012, D'Amour et al., 2021], have surprisingly received limited attention in the policy learning literature. Among the mentioned works on policy learning, only [Chakraborty and Moodie, 2013] mentions that methods for handling positivity violations in multi-stage settings are underdeveloped.

This lack of focus on positivity violations is also evident in the available software implementations. Notably, packages such as DynTxRegime (R) [Holloway et al., 2022], DTRlearn2 (R) [Chen et al., 2020], and EconML (Python) [Battocchi et al., 2019] do not provide methods for handling practical positivity violations, apart from inverse probability weight truncation. The R software package polle [Nordland and Holst, 2022] was specifically developed to tackle the extensive practical positivity violations observed in the application discussed in this work.

3. Concepts

In this section, we present a general framework for corrective maintenance and repair processes and introduce the notation required to define a policy learning algorithm. This algorithm addresses the unique challenges associated with utilizing historical data where the decisions have not been randomized. Our primary objective is not to provide an exhaustive description of the methods employed, but rather to introduce the essential concepts necessary for understanding the model components. We closely follow the notation established by [Nordland and Holst, 2023], where a comprehensive description of the methods can be found.

Let us consider a scenario where we have a dataset consisting of N independent units. This means that the condition and usage of each unit do not impact the other assets. The data is collected over a fixed follow-up period, denoted as $[0, T]$. During this period, each unit is assigned several work orders, which outline different maintenance and repair tasks along with their associated costs. These work orders can either be approved or rejected. If a work order is approved, all the specified tasks are carried out. In many cases, if a work order is not approved, the unit will no longer be operational and will be sold. As long as the unit remains in working condition, it generates revenue for the business. At the end of the follow-up period, we reward each operational unit with an in-service cash premium.

Our objective is to maximize the expected profit of the unit fleet throughout the follow-up period, considering all the costs and revenue streams associated with each unit. It's important to note that the optimization task described here is corrective or reactive, meaning that we cannot control the timing of the work orders. Our decision-making is limited to approving or rejecting each work order. Thus, we want to estimate a policy, a set of rules, which given the individual history of the unit returns the optimal decision in terms of the long term expected profit.

At this point, we are ready to introduce the formal notation for the process. We begin by considering a fixed-time process, as illustrated in Figure 1. Each process consists of a fixed number K of work orders, which define $K+1$ stages of the process.

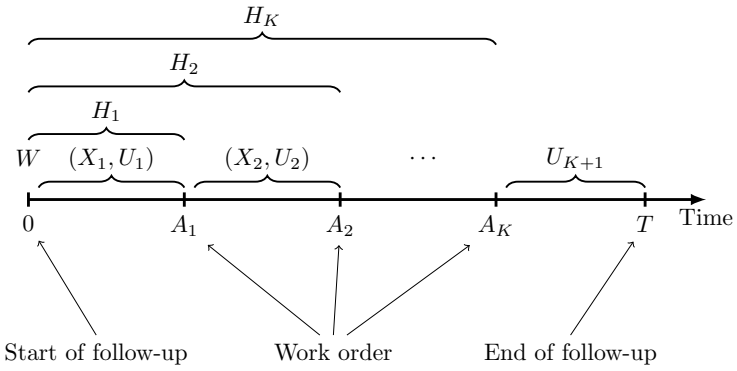


Fig. 1. Maintenance process with fixed time intervals. W represents the baseline variables, A_k, $k = 1, 2, ...$, the action variables, and H_k the available history prior to the decision. After the action is made, the state variables X_{k+1} and the reward U_{k+1} are updated.

We define $W \in \mathcal{W}$ as the baseline variable containing the unit specifications. At each stage $k \in \{1, \ldots, K\}$, we use $A_k \in \{0, 1\}$ to represent the action taken, where $A_k = 1$ corresponds to approving the work order at stage k. To summarize the data collected up to stage 1, we use S_1. For $k \in \{2, \ldots, K\}$, S_k represents a summary of the data collected between stage $k-1$ and k. Similarly, S_{K+1} summarizes the data collected after stage K. To simplify notation, we let $S_k = (X_k, U_k)$, where U_k denotes the kth reward, and X_k serves as a stage variable summarizing the work order at the given stage and other relevant information. For convenience, we define $X_{K+1} = \emptyset$. Thus, using the implied ordering, the observed process can be written as

$$O = (W, S_1, A_1, S_2, A_2, \ldots, S_K, A_K, S_{K+1}).$$

For $k \in \{1, \ldots, K+1\}$, let $\overline{S}_k = (S_1, \ldots, S_k)$, $\overline{A}_k = (A_1, \ldots, A_k)$ and $H_k = (W, \overline{S}_k, \overline{A}_{k-1}) \in \mathcal{H}_k$ define the full history at stage k where $A_0 = A_{K+1} = \emptyset$. Finally, the utility, specifically the profit, is calculated as the sum of the rewards.

$$U = \sum_{k=1}^{K+1} U_k.$$

3.1. Policies

A policy is a set of rules $d = (d_1, ..., d_K)$, where $d_k : \mathcal{V}_k \mapsto \mathcal{A}$ assigns an action in stage k given a summary V_k of the history H_k. We informally denote an observation where all actions are taken according to policy d as O^d. For the optimal policy d_0 it holds that $\mathbb{E}[U^d] \leq \mathbb{E}[U^{d_0}]$ for any other policy d restricted to the same input. If V_r is a function of V_k for any $r \in \{1, \dots, k-1\}$, then the optimal policy can be expressed using the (optimal) *blip functions*. For any actions $a = (a_1, ..., a_{K-1})$ recursively define

$$
\begin{aligned}
B_{0,K}(\overline{a}_{K-1}, v_K) &= \mathbb{E}\left[U^{(\overline{a}_{K-1}, a_K=1)} - U^{(\overline{a}_{K-1}, a_K=0)} \middle| V_K^{\overline{a}_{K-1}} = v_K\right], \\
d_{0,K}(\overline{a}_{K-1}, v_K) &= I\left\{B_{0,K}(\overline{a}_{K-1}, v_K) > 0\right\}, \\
B_{0,k}(\overline{a}_{k-1}, v_k) &= \mathbb{E}\left[U^{(\overline{a}_{k-1}, a_k=1, \underline{d}_{0,k+1})} - U^{(\overline{a}_{k-1}, a_k=0, \underline{d}_{0,k+1})} \middle| V_k^{\overline{a}_{k-1}} = v_k\right] \quad k \in \{1, ..., K-1\}, \\
d_{0,k}(\overline{a}_{k-1}, v_k) &= I\left\{B_{0,k}(\overline{a}_{k-1}, v_k) > 0\right\} \quad k \in \{1, ..., K-1\},
\end{aligned}
$$

where we use the notation $\underline{d}_k = (d_k, ..., d_K)$, and I denotes the indicator function. Importantly, the above statement remains true if we, instead of assuming that V_r is a function of V_k for any $r \in \{1, \dots, k-1\}$, assume that

$$
\mathbb{E}\left[U^a \middle| V_1, V_2^{a_1}, \dots, V_k^{\overline{a}_{k-1}}\right] = \mathbb{E}\left[U^a \middle| V_k^{\overline{a}_{k-1}}\right]. \tag{1}
$$

So when we, in practice, design the summary V_k, we do not necessarily need to compound all previous summaries. We just need to capture all (or most) of the contained information relevant for the utility.

The blips serve to quantify the difference in value between the two actions when all subsequent actions are optimal. They provide insight into the potential gain or loss resulting from choosing one action over the other, taking into account the optimality of future actions. The blips are causal functions that may not be directly identifiable from the historical data. To identify the blips, we rely on two key structural assumptions: positivity and sequential randomization [Robins, 1986]. The positivity assumption states that there must be a positive probability of choosing each action at each stage, given the history. This assumption ensures that all possible actions have a chance of being selected, allowing for a comprehensive exploration of the action space. Define the g-functions as

$$
g_{0,k}(h_k, a_k) = \mathbb{P}(A_k = a_k \mid H_k = h_k).
$$

Under positivity it holds that $g_{0,k}(H_k, a_k) > 0$ almost surely for any a_k. In practice, violations of the positivity condition, or even near positivity violations, are a cause for concern. We will discuss this issue later on.

When the guidelines for approving work orders are strict, the historical actions may exhibit limited variability. Therefore, it is not feasible to estimate the overall optimal policy. Instead, we target the optimal realistic policy that maintains positivity at a specified level. We will discuss this in more detail later in this section.

The sequential randomization assumption establishes a connection between the historical data and the causal parameter. The assumption states that the future potential outcomes are independent of the observed action given the history at a particular stage:

$$
\{S_{k+1}^{\overline{a}_k}, \dots, S_{K+1}^{\overline{a}_K}\} \perp A_k \middle| W, \overline{S}_k, \overline{A}_{k-1} = \overline{a}_{k-1} \quad \forall k \in \{1, \dots, K\}.
$$

The assumption holds when we effectively capture all the information that has influenced the decision-making process. However, if we fail to account for confounding factors that influence both the historical actions and rewards, the estimated optimal policy can be highly misleading and may even have harmful effects compared to status quo. It is crucial to carefully consider and address these potential confounding factors to ensure the reliability and effectiveness of the estimated policy.

Under positivity and sequential randomization is possible to construct a doubly robust loss function for the blips. For a given feasible policy d define the doubly robust blip score as

$$
\begin{aligned}
&Z_k(\underline{d}_{k+1}, g, Q^{\underline{d}_{k+1}})(O) \\
&\quad = Q_k^{\underline{d}_{k+1}}(H_k, 1) - Q_k^{\underline{d}_{k+1}}(H_k, 0) \\
&\quad + \sum_{r=k}^{K} \left\{ \frac{2A_k - 1}{g_k(H_k, A_k)} \prod_{j=k+1}^{r} \frac{I\{A_j = d_j(H_j)\}}{g_j(H_j, A_j)} \right\} \left\{ Q_{r+1}^{\underline{d}_{r+2}}(H_{r+1}, d_{r+1}(H_{r+1})) - Q_r^{\underline{d}_{r+1}}(H_r, A_r) \right\},
\end{aligned} \tag{2}
$$

where $\prod_{k+1}^{k} x_k = 1$ and where the Q-functions are recursively defined as

$$Q_{0,k}^{\underline{d}_{k+1}}(h_k, a_k) = \mathbb{E}\left[Q_{0,k+1}^{\underline{d}_{k+2}}(H_{k+1}, d_{k+1}(H_{k+1})) \mid H_k = h_k, A_k = a_k\right], \quad k \in \{1, \ldots, K\},$$

and $Q_{K+1}^{\underline{d}_{K+2}}(H_{K+1}, d_{K+1}(H_{K+1})) = U$. The blip score is doubly robust in the sense that if either $g = g_0$ or $Q^{d_0} = Q_0^{d_0}$ then

$$\mathbb{E}\left[Z_k(\underline{d}_{0,k+1}, g, Q^{\underline{d}_{0,k+1}})(O)\middle|\overline{A}_{k-1}, V_k\right] = B_{0,k}(\overline{A}_{k-1}, V_k),$$

Thus, a valid loss function for the blip at stage k is given by

$$L_k(B_k)(\underline{d}_{0,k+1}, g, Q^{\underline{d}_{0,k+1}})(O) = \left(Z_k(\underline{d}_{0,k+1}, g, Q^{\underline{d}_{0,k+1}})(O) - B_k(\overline{A}_{k-1}, V_k)\right)^2, \tag{3}$$

since $\mathbb{E}[L_k(B_k)(\underline{d}_{0,k+1}, g_0, Q_0^{\underline{d}_{0,k+1}})]$ is minimized in the true blip $B_{0,k}$. The above results inspire a recursive regression type estimator for the blips and, consequently, the optimal policy [Luedtke and van der Laan, 2016].

When it comes to non-parametric estimation of functions, including the blips, it is worth noting that pointwise evaluation of these functions often lacks pathwise differentiability [Luedtke and Chung, 2023]. This means that there is no asymptotically root-n estimator available for these functions. As a result, we cannot provide performance guarantees for the estimated policy based on these non-parametric estimators.

However, in the case of single-stage problems, it has been demonstrated that a reduction in the complexity of the candidate set of policies to fixed linear models, decision trees or reproducing kernel Hilbert spaces is enough to establish performance guarantees towards the best performing policy within the set [Luedtke et al., 2020, Luedtke and Chung, 2023]. To our knowledge, the results mentioned above have not yet been extended to multi-stage problems. However, we believe it is reasonable to apply the same methodology to the multi-stage case.

Reducing the complexity of the estimated policy may offer several other benefits. Firstly, it can make the policy easier to interpret. This is especially the case for low-depth policy decision trees and linear models with a small number of variables. However, in practice, we see a trade-off between the performance of the policy and interpretability. Secondly, a less complex policy can ease implementation in real-world scenarios. In some cases, it may be costly or time-consuming to collect the necessary data for a complex policy, or the decision-maker may not have immediate online access to compute the policy recommendations. By simplifying the policy, we can make it more feasible to implement and apply it in a practical setting.

3.2. Near-positivity violations

At this point, the recursive policy estimation approach outlined does not prevent near positivity violations. The use of inverse probability weights in the doubly robust blip score can lead to rare instances of numerically large weights, which in turn can cause instability in the regression for the blip. Although the estimated Q-functions can contribute to stabilizing the scores, the consistency of these functions may be compromised when there is limited variability in the observed actions. Consequently, there is a reliance on uncertain extrapolation when attempting to estimate and generalize the Q-functions under such circumstances.

To address this issue, we suggest a solution that involves constraining the set of feasible actions taken by the policy. By introducing an action probability threshold $\alpha > 0$, we can modify the estimated policy recursively as follows:

$$d_k^\alpha(h_k) = I\{g_k(h_k, 1) \in (\alpha, 1-\alpha)\} I\{B_k^\alpha(\overline{a}_{k-1}, v_k) > 0\} + I\{g_k(h_k, 1) \in (1-\alpha, 1)\}.$$

Here, B_k^α represents the blip associated with subsequently following the optimal realistic policy at the α level. Ensuring positivity protection is crucial for applications where positivity violations are a real concern. The importance of this protection will be emphasized in our simulation, see Section 4. The downside of modifying the estimated policy in this fashion is that the new policy depends on the existing action model via the g-functions. If collecting the input for the existing action model poses issues for implementing the protected policy, we propose fitting a new simplified g-function based on the policy input V_k. This approximated g-function can then be utilized to provide the necessary protection.

3.3. Stochastic number of decision stages

Up to this point, we have only considered an maintenance and repair process with a fixed number of stages. However, in our application, the timing between work orders varies within the follow-up period, resulting in a stochastic number of work order stages K (as depicted in Figure 2). If the maximum number of stages is bounded, we can still apply the outlined methodology by augmenting each observation to have the same number of work order stages [Goldberg and Kosorok, 2012]. It is worth noting that at stage k, the blip score, and consequently the estimated policy, only depend on the observations with at least k stages.

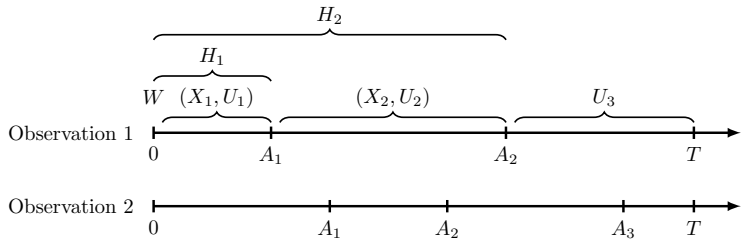


Fig. 2. Maintenance process in continuous time were the work orders occurs randomly over time.

4. Equipment Repair and Maintenance Simulation Study

The purpose of this section is twofold. First, we aim to compare the performance of standard Q-learning and realistic doubly robust blip-learning in scenarios involving multiple stages with limited variation in the actions, which can lead to positivity violations. Second, we aim to verify the consistency and asymptotics of the estimated policy values. Both aspects are crucial for the application of policy learning, especially in the context of equipment maintenance and repair optimization.

The simulation model is based on a simplified fit of our actual application concerning reefer equipment maintenance as described in Section 5. This approach allows us to closely resemble the data structure, including the stochastic nature of the number of work orders, the action model, and the utility measure.

To simulate the data, we employ a Markov structural equation model (SEM) that generates the data sequentially for each stage. Table 1 provides a description of the baseline, stage-specific variables, and endline variables in the model. Additionally, Figure 3 presents a graphical representation of the SEM. Each equipment unit is followed from the age of 5 years to the age of 16.5 years, resulting in a follow-up period in weeks ranging from 261 to 860. Each stage of the process is repeated as long as the work order is approved (action 1) and the unit has not reached an age of more than 860 weeks. Otherwise the process is stopped. For convenience, we set all subsequent stage variables to 0. The 16th work order will always be rejected, limiting the maximum number of work orders to 16. If a unit reaches week 860, the number of moves and the additional cost only depends on the time between the age at the previous stage and week 860. It is important to note that we only sample observations that have at least one work order within the follow-up period.

The time variable model is based on a Weibull proportional hazard regression, implemented in the R package `mets` [Scheike et al., 2014]. The remaining variable models are based on penalized regression splines and tensor product splines, implemented in the R package `mgcv` [Wood, 2017]. The source code and further documentation for the simulation are available in the R package `emrsim`[1].

The last missing component of the simulation model is to define the rewards as a function of the variables in the SEM. Naturally, we define the rewards in terms of the profits associated with the reefer equipment between each stage. For this purpose we set a fixed cash equivalent of 700 for each move the reefer unit makes. The reward associated with stage $k \leq K$ (U_k) is now given by

$$\text{reward}(k) = \Big[\text{moves}(k) \times 700\Big] - \text{additional cost}(k) - \Big[\text{action}(k-1) \times \text{work order cost}(k-1)\Big].$$

And at stage $K+1$ the final reward (U_{K+1}) is given by

$$\text{reward}(K+1) = \begin{cases} \text{in service premium} - \text{requisition cost} + \Big[\text{moves}(K+1) \times 700\Big] - \text{additional cost}(K+1), & \text{if age}(K+1) > 860 \\ \text{sales price} - \text{requisition cost}, & \text{if age}(K+1) \leq 860 \end{cases}$$

As the requisition cost is fixed it will have no effect on the policy learning. We only include the requisition cost to centralize the value and make the rewards comparable to the real application. The baseline variables, W, are defined by the type of box and unit. The state variables, X_k, includes time, age, costs and moves.

As described, the number of work orders within the follow-up period is stochastic due to the varying timing between each stage. Figure 4 displays the number of approved and rejected work orders at each stage for a random sample of 20,000 observations. Due to the construction of the process, only a few observations experience a large number of work orders within the follow-up period. Consequently, the estimation of the blip-functions for the late stages solely relies on a small

[1] `https://github.com/kkholst/emrsim`

Table 1. Variables in the Structural Equation Model (SEM) for simulating reefer maintenance and repair data.

Baseline variables			
Name	Description	Type	Conditional Distribution
Baseline age	Age of the reefer at baseline	Continuous	Fixed (5 years/261 weeks)
Box	Type of the reefer box	Binary	Bernoulli
Unit	Type of the refrigeration unit	Binary	Bernoulli
Stage variables			
Time (k)	Time since the previous stage.	Continuous: $[0, \infty)$	Weibull (proportional hazard)
Age (k)	Age of the reefer.	Continuous: $[0, \infty)$	
Moves (k)	The number of moves made by the reefer since the previous stage.	Discrete: $[0, \infty)$	Poisson (log link)
Additional cost (k)	Additional maintenance costs accumulated since the previous stage.	Continuous	Gaussian
Work order cost (k)	Costs associated with approving the work order at the given stage.	Continuous	Gaussian
Action (k)	Decision on whether to approve the work order.	Binary	Bernoulli (logit link)
Endline variables			
Sales price	Sales price of the reefer if the reefer is scrapped within the follow-up period.	Continuous	Gaussian
In service premium	Cash premium if the reefer is in working condition at the end of the follow-up period.	Continuous	Fixed (4000)
Requisition cost	Price for a new reefer.	Continuous	Fixed (14000)

sub-sample of the data, which may result in erratic behavior of the estimated policy. To address this, we limit the maximum number of stages to 8 by focusing on partial policies that only intervene on the first 8 stages. The remaining stages will continue to be determined by the existing action model. This ensures more reliable estimation and avoids potential issues caused by the lack of data in the later stages.

To illustrate the limited support of the g-function, Figure 5 depicts the contours of the propensity, i.e., the probability of approving a work order, as a function of age and work order cost. For situations with low age and low work order cost, it is unrealistic to observe a rejection of the work order. Conversely, for high age and high work order cost, it is unrealistic to observe an approval of the work order. As a result, when estimating a realistic optimal policy at a given α-level, e.g. $\alpha = 0.1$, the optimization will only focus on a narrow band of the stage variable space. Over the remaining state variable space, the most probable action is selected to avoid positivity violations.

4.1. Nuisance model estimation

The blip policy learning method requires estimation of the nuisance g and Q-functions. We do not fit the g-functions separately for each stage. Instead, we fit a single g-function across all stages at once taking advantage of the known Markov structure for the decision process, i.e., that the g-function is the same for every stage:

$$g_{0,k}(H_k, A_k) = g_0(Z_k(H_k), A_k) \qquad \forall k \in \{1, ..., K\}.$$

In this case the subset Z_k contains the age and the work order cost at stage k. The g-function is modeled using a generalized additive model with integrated smoothness estimation [Wood, 2017] as implemented in the R package `mgcv`. The model includes a tensor product smoother on the age and work order cost. The model is the same model used to simulate the actions in the SEM and as such it represents the true underlying model.

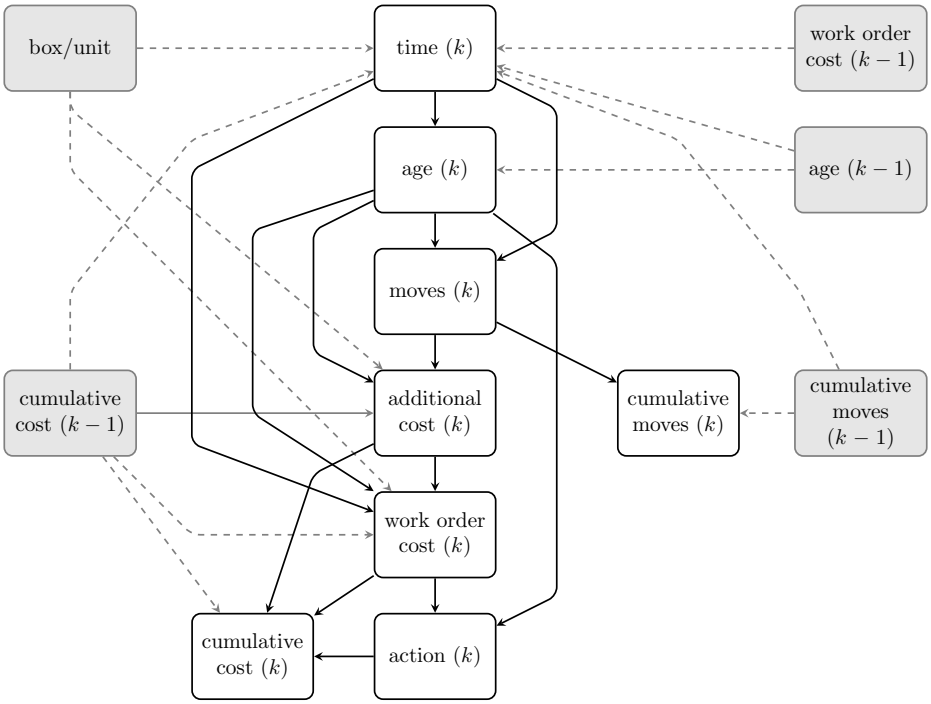


Fig. 3. Graph of the structural equation model at stage k. The model is used to simulate data similar to the application in Section 5

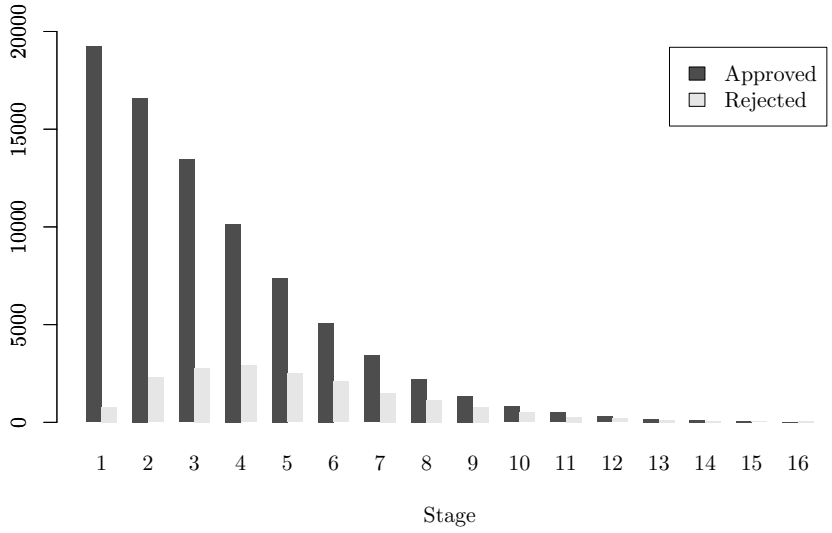


Fig. 4. Count of approved and rejected work orders for each stage for 20,000 simulated observations.

The Q-functions are not fitted using the full available history at each stage. Instead, only the variables associated with each stage, along with the cumulative costs and the cumulative number of moves, are provided. This simplification is justified if it holds that

$$Q_{0,k}^{d_{k+1}}(h_k, a_k) = Q_{0,k}^{d_{k+1}}(l_k, a_k) \qquad \forall k \in \{1, ..., K\}, \tag{4}$$

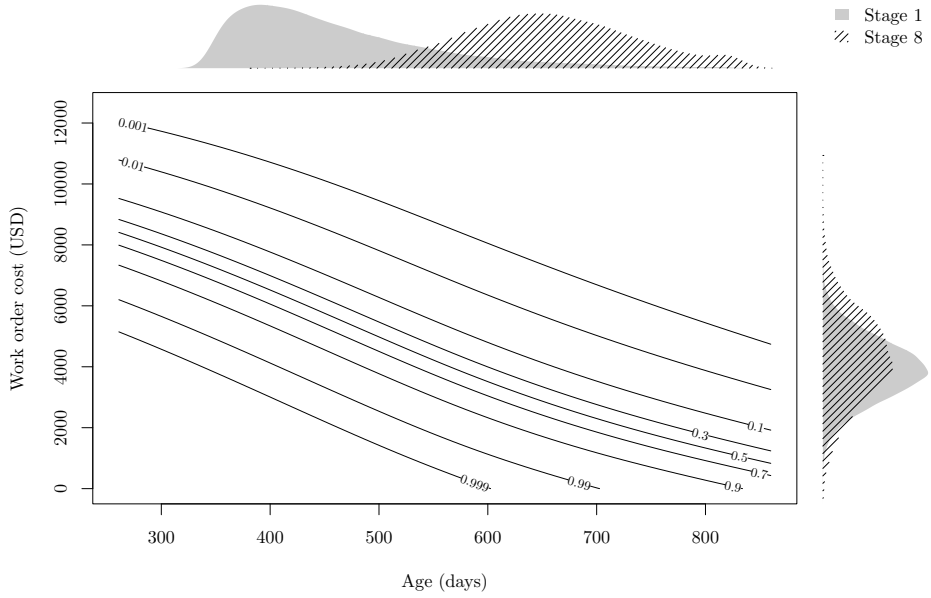


Fig. 5. The true g-model propensity for approving a work order for varying age and work order cost. The remaining variables (box,unit, and region) are kept fixed.

where l_k is the considered summary of h_k. In practice, the Q-functions are fitted using the R package `SuperLearner` [Polley et al., 2021]. The model creates a weighted average of an ensemble of models based on the cross-validated risk (10 folds). The ensemble includes linear regression models, generalized additive models with discretization, lasso models with natural cubic spline bases, and a regularized gradient boosting tree model. [Chen et al., 2023]. All models, except the gradient boosting tree model, include all possible action interactions.

4.2. Policy learning & evaluation

We use the R package `polle` [Nordland and Holst, 2022] and its `policy_learn` function to estimate the blip functions and the corresponding policy. For doubly robust evaluation, we calculate scores similar to Equation (2) and estimate the value by taking the empirical mean of the scores. The centralized scores are actually estimates of the influence function terms for the associated value estimator, making it easy to construct Wald-type confidence intervals. The policy evaluation method is implemented in the `polle` package through the `policy_eval` function, see [Nordland and Holst, 2022] for more details.

In addition to the doubly robust value estimator, we can also estimate the policy value using inverse probability weighting (IPW) or the empirical mean of the fitted stage one Q-function. The former we refer to as the outcome regression (OR) estimator. Ideally, both the policy learning and evaluation procedures would involve cross-fitting the g and Q-functions. However, for computational simplicity, we opt to use the in-sample fitted nuisance functions.

In practice, selecting a suitable regression model for the blip functions is challenging, especially when the number of observations in each of the considered eight stages varies as significantly as in this case. The later stages may not support as flexible a blip function as the earlier stages. To address this, we choose to use a super learner for the blip functions based on the doubly robust loss function in Equation (3). In the ensemble we include simple linear models and more flexible linear models with unit interaction terms, spline basis expansions of continuous variables, and product interaction terms. This approach was first suggested by [Luedtke and van der Laan, 2016] and further studied by [Montoya et al., 2022].

The blip policy learning method is benchmarked against regular Q-learning with the same nuisance model specifications. For reference, we also include Q-learning based on a linear model and a action stratum conditional mean model. We apply realistic versions of all of the policy learners. The simulation results are presented in Table 2.

Firstly, standard Q-learning based on the flexible nuisance models outperforms the blip policy learner. This result may not be that surprising. If the estimated Q-functions are consistent and have low variability, the associated policy approximates the optimal policy well. The low bias and RMSE of the outcome regression estimator indeed give an indication that the fits of the Q-functions are good. Notably, although Q-learning in this case is insensitive to positivity violations, adding some positivity protection does not negatively impact the learner's performance. And importantly, increasing the protection level reduces the bias of the doubly robust and outcome regression value estimators by half. In comparison, the

pl	qm	gm	alpha	value	SD (value)	bias (DR)	RMSE (DR)	coverage (DR)	bias (IPW)	RMSE (IPW)	bias (OR)	RMSE (OR)
blip	mean	gam	0.000	994	645	14472	91683	0.43	6534	43123	-410	467
blip	mean	gam	0.010	1178	74	140	159	0.44	56	257	-419	453
blip	mean	gam	0.020	1212	53	98	121	0.65	79	129	-376	408
blip	mean	gam	0.030	1200	54	74	99	0.67	35	132	-387	417
blip	mean	gam	0.040	1228	36	53	79	0.76	45	99	-351	372
blip	mean	gam	0.050	1222	52	64	85	0.80	45	104	-360	382
blip	mean	gam	0.075	1222	37	33	60	0.93	21	75	-337	364
blip	mean	gam	0.100	1214	35	23	61	0.94	8	80	-325	339
blip	sl	gam	0.000	1160	181	287	403	0.81	99	1635	13	52
blip	sl	gam	0.010	1279	40	86	117	0.75	143	223	11	65
blip	sl	gam	0.020	1283	39	68	104	0.73	46	251	11	59
blip	sl	gam	0.030	1285	34	54	84	0.82	50	138	2	50
blip	sl	gam	0.040	1282	29	60	86	0.80	87	141	11	54
blip	sl	gam	0.050	1292	27	37	75	0.89	24	217	-2	50
blip	sl	gam	0.075	1281	34	21	58	0.93	59	111	-13	58
blip	sl	gam	0.100	1255	34	24	49	0.94	43	108	-13	46
ql	glm	glm	0.000	1241	20	-287	497	0.88	271	691	361	365
ql	glm	glm	0.010	1257	20	-63	149	0.92	227	484	334	340
ql	glm	glm	0.020	1264	20	-32	102	0.91	153	256	296	301
ql	glm	glm	0.030	1263	20	-23	81	0.90	144	230	279	284
ql	glm	glm	0.040	1262	21	-4	63	0.95	127	173	262	267
ql	glm	glm	0.050	1257	18	2	59	0.96	115	182	260	266
ql	glm	glm	0.075	1251	22	-7	54	0.94	81	149	216	222
ql	glm	glm	0.100	1238	20	-0	60	0.89	65	104	193	202
ql	sl	gam	0.000	1346	22	49	71	0.87	54	203	39	61
ql	sl	gam	0.010	1354	20	41	77	0.82	61	160	38	68
ql	sl	gam	0.020	1350	22	26	64	0.88	47	153	23	59
ql	sl	gam	0.030	1347	19	28	65	0.89	53	138	22	57
ql	sl	gam	0.040	1346	19	22	57	0.92	73	132	15	52
ql	sl	gam	0.050	1341	20	26	69	0.84	61	125	15	56
ql	sl	gam	0.075	1322	19	20	61	0.89	61	120	8	54
ql	sl	gam	0.100	1298	19	14	52	0.93	29	91	1	44

Table 2. Policy learning and evaluation simulation. The 'pl' column specifies the policy learning method, where 'ql' refers to Q-learning and 'blip' refers to blip-learning. The 'qm' and 'gm' columns indicate the choice of models for the Q-functions and g-functions, respectively. 'glm' represents a generalized linear model, 'sl' represents Super Learner, and 'gam' represents a generalized additive model with integrated smoothness estimation. The parameter α defines the considered realistic action set, with $\alpha = 0$ implying no restriction on the policy learner. Each row is based on 100 replications with a sample size of 20,000. For each estimated policy, the true value is approximated by Monte Carlo sampling with 100,000 observations under the policy. The value under the existing observed action model is 729.

Q-learner and outcome regression value estimator based a simple linear model ('qm':'glm') is overly optimistic as evident from the value bias. Despite this, both the estimated policy and the doubly robust value estimator perform decently when we add some positivity protection.

The performance of the blip policy learner is decent considering the induced simplicity of the fitted policy. However, it is evident that this policy learner is more susceptible to positivity violations. When no positivity protection is added to the policy ($\alpha = 0$), the inverse probability weighting terms in the doubly robust blip scores exhibit high variability, resulting in unstable policies and high variation in value. Restricting the policy learner by increasing the α-level enhances performance and stability considerably. In our simulation, the highest performance is achieved at $\alpha = 0.05$. Further increasing the α-level improves the consistency of the value estimator, albeit at the expense of the policy performance. To showcase the double robustness property of the blip learner we include the case where we let the Q-function be fitted as the empirical mean for each action ('qm':'mean'). In this case the blip learner will solely rely on the g-functions for consistency. It is clear from the results that this learner is even more susceptible to positivity violations. However, with some protection, the learner still manages to estimate a reasonably performing policy.

Overall, the simulation study suggests that adding positivity protection to any policy learner under consideration is beneficial. The additional protection may contribute to improved policy performance. What is clear is that our ability to evaluate policy performance is improved. However, a question arises: how do we determine the optimal level of protection that strikes a balance between policy performance and evaluation? It should be noted that the number of stages in the analysis plays a crucial role in determining the appropriate level of protection.

From a practical standpoint, we suggest to monitor the changes in the estimated doubly robust value and its associated standard error as the positivity protection level is increased. Although the value estimator is doubly robust, its coverage depends on the consistency of both the Q and g-functions. Therefore, it is important to also monitor the outcome regression and inverse probability weighting estimators. We conjecture that notable deviations in the value estimates and the standard error justifies increasing the protection level. To support this, we present the value estimates and standard error for a single run of the blip learner policy evaluation simulation in Figure 6. Although the outcome regression estimate remains stable even for $\alpha = 0$, the inverse probability weighting estimator clearly indicates practical positivity violations. This issue also

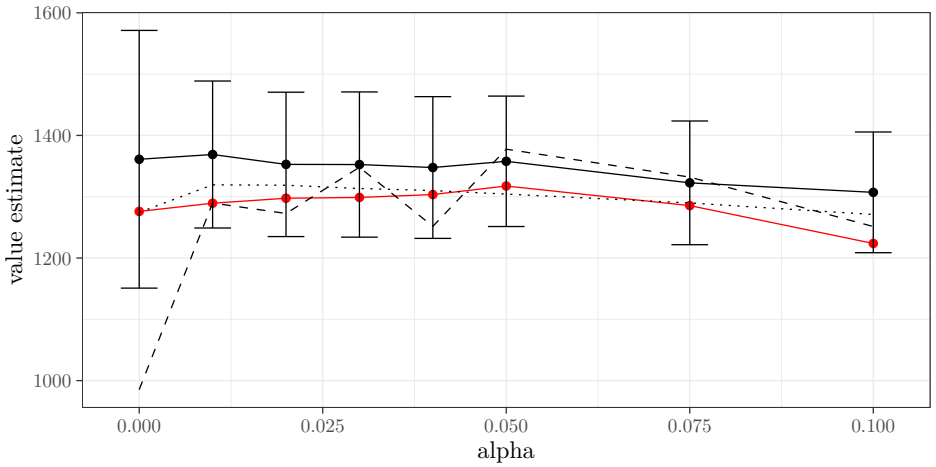


Fig. 6. Plot of the estimated policy value for changing values of the protection level α. The results are based on a single run of the blip learner policy evaluation. The Q-functions are fitted using a super learner, and the g-function is fitted using a generalized additive model. The solid black line and points represent the doubly robust value estimates. The true value of the fitted policy is represented by the red line and points. The dashed line represents the inverse probability weighting estimates, and the dotted line represents the outcome regression estimates.

affects the estimated standard error of the doubly robust estimator. Protection levels between 0.01 and 0.05 show similar estimated performance. In this case, we argue that it is sensible to select the highest protection level.

5. Application - Reefer Maintenance

5.1. Setting & data

Maersk owns a large fleet of more than 300,000 reefers used primarily to transport chilled or frozen perishable cargo. Reefers are considerably more expensive to acquire than dry containers and with an intended lifetime between 12 and 20 years, the fleet of reefers represent a substantial long term investment for the company. As with all containers, reefers have a high wear and tear level, and a large amount of resources must be spend to maintain the fleet. Our aim is to leverage the available historic data to identify maintenance policies that will maximize the value of each reefer across its lifetime. Because of the operation scale, even small improvements in the fleet management can greatly impact the business.

Every cleaning, inspection, repair, or maintenance task associated with a reefer is recorded as a work order. Each work order consists of a list of items describing the individual tasks and the required materials, along with the estimated cost of each item. The items are categorized as either box items or refrigeration unit items. Different repair shops may handle various items within a work order. Additionally, each work order is associated with a mode that indicates whether the reefer is laden (loaded with cargo) or empty.

Work orders that involve routine tasks like pre-trip inspections and cleanings are automatically approved. We exclude such work orders from the decision process. Thus, we only include work orders with a box or refrigeration unit cost estimate above a given threshold. All of these work order requires approval from an equipment manager. The manager is supported by operational guidelines that take into account factors such as the age of the reefer, the region it is operating in, and specific cost limits for both box and refrigeration unit repairs. Regional differences occur because labor costs and demand for empty reefers varies across different regions. Another important consideration for approving or rejecting a work order is to determine the party liable for the damage, whether it's the customer or a third party. In cases where a work order is rejected, the reefer is typically sold off or scrapped.

The data set is based on all Maersk owned 40ft steel reefers manufactured in 2000 and 2001. The data set contains information on the specifications of the reefers and detailed information on the usage and location histories as well as all work orders. Lastly, the data set contains information on sales and scrap prices for reefers that have been sold. The reefers are followed 11.5 years from the age of 5. Thus, the subsequent analysis is conditional on reefer surviving the first 5 years. Furthermore, we exclude 179 reefers with no work orders within the follow-up period. These reefers will not be affected by any intervention on the work order process and can thus safely be excluded for our purposes. In total the cohort consists of 17,704 reefers. Figure 7 (a) displays the number of active reefers in the cohort over time. After 10 years, the disposal rate of

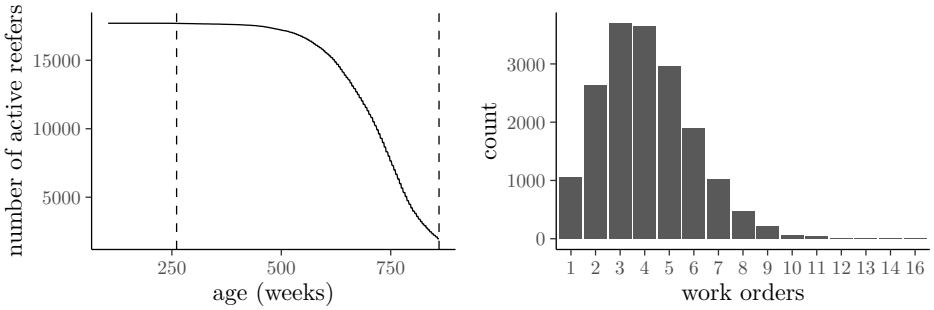


Fig. 7. (a): Number of active reefers over time in the cohort. The dashed lines mark the follow-up period. (b): Number of work orders within the follow-up period for the reefers in the cohort.

the reefers increases rapidly. As expected, the reefers have a stochastic number of work orders within the follow-up period, see Figure 7 (b).

5.2. Defining the utility

Defining a utility measure is crucial for quantifying the potential impact of implementing a given policy. The utility measure should accurately capture the priorities and objectives of the business. In the context of reefers, their utility is determined by revenue generation through service provisions and reefer sales, as well as the costs associated with maintenance, repairs, and the initial purchase of the reefer.

To establish a utility measure, we assign a fixed cash equivalent for each type of service provided by the reefer. We differentiate between live trips, where the refrigeration unit is in use, and non-live trips. This allows us to calculate the revenue stream generated by each reefer within the follow-up period. As mentioned, the cost stream is defined as the expenses associated with maintenance and repairs of the reefer during the same period.

The timing of the revenue and costs is important for the business, as it influences the availability of capital for alternative investments. To account for this aspect, we use the (US) inflation curve to discount the revenue and cost streams such that early costs are penalized, and early revenues are rewarded.

If a reefer is sold within the follow-up period we add the selling price to revenue stream and discount it according to the sales date. If a reefer is not sold within the follow-up period, a cash premium equivalent to the (discounted) sales price of a reefer in working condition is added to the revenue stream at the end of the follow-up period.

The individual reefer purchase prices are not available for this application. Thus, we cannot account for the potential differences in purchase prices. For this simple reason the purchase prices are not included in the cost stream. Finally, we assume that the business immediately wants to replace a reefer that have been sold to maintain the fleet capacity. Thus a fixed requisition cost is added to the cost stream at the time of the sale or at the end of the follow-up period. Importantly, the requisition cost is then discounted accordingly.

The individual rewards can now easily be defined as the discounted profit generated between each stage in the work order process. By definition, the utility is defined as the overall discounted profit.

5.3. Nuisance model estimation

The information available to the equipment manager for the approval of a specific work order is recorded in a centralized operational system. Importantly, the equipment manager does not have access to the full maintenance history of the reefer. Therefore, it is reasonable to assume that the action model is independent of the stage number. The information available to the equipment manager can be represented as a subset Z_k of the full history H_k. As a result, we can simplify the g-functions as follows:

$$g_{0,k}(H_k, A_k) = g_0(Z_k(H_k), A_k) \qquad \forall k \in \{1, ..., K\}.$$

Specifically, Z_k contains information on the cost estimates for the box and the refrigeration unit, age, location, and specifications. Furthermore, we include 14 binary system notifications flagging various conditions. We reasonably assume that Z_k is the minimal set of variables required to ensure sequential randomization at stage k.

Due to the Markov structure of the g-function, we stack the data across all 8 stages. The g-function is then cross-fitted using 20 folds. Within each training fold the g-function is fitted using the super learner algorithm as described in Section 4. The ensemble includes various logistic regression models, generalized additive models with discretization and gradient

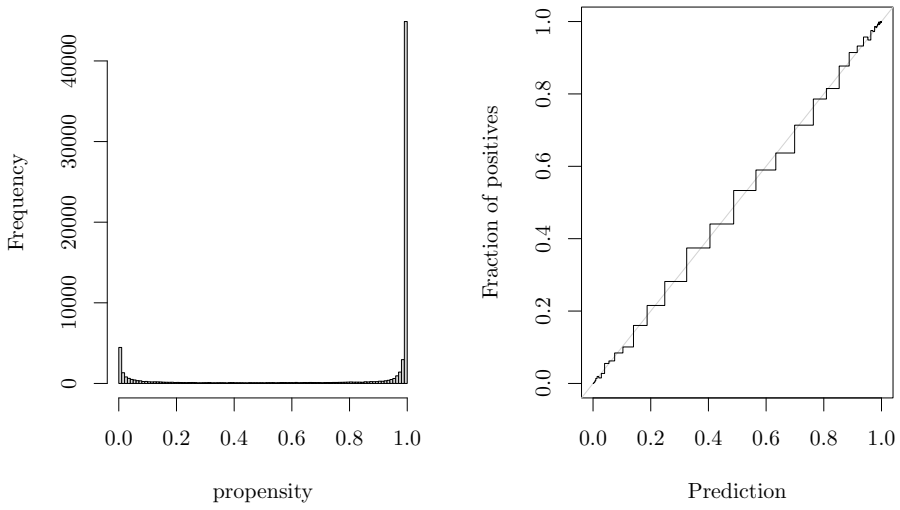


Fig. 8. Histogram and calibration plot of the cross-fitted g-function values.

boosting tree models. A histogram of the cross-fitted g-function propensities, i.e., $g(H_k, a_k = 1)$, and a bin calibration plot are displayed in Figure 8.

The equipment manager typically adheres to the implemented guidelines, where work orders with cost estimates below specified limits are approved, and those exceeding the limit are not. However, deviations from these guidelines occur due to variations in regional demand for reefers and repair shop capacity. We assume that these variations are completely random in the sense that they do no act as confounders for the future rewards. Anyhow, we expect the cross-fitted values of the g-function to be heavily skewed towards 0 and 1, which is indeed the case as seen in Figure 8. Additionally, it is evident that the cross-fitted values of the g-function are well calibrated. In our practical experience, when employing flexible models for the g-functions, we find that the cross-fitting procedure has a positive impact on calibration.

Theoretically, the compounded history $\{W, Z_1, A_1, \ldots, Z_{k-1}, A_{k-1}, Z_k\}$ is the minimal set of variables that can be used as input for the Q-function at stage k. However, for practical purposes, we first aim to summarize the full history such that Equation (4) holds. Specifically, we construct the the Q-function input to include the g-function input, Z_k, and the cumulative work order costs associated with the box and the refrigeration unit. The previous actions are omitted from the summary since they are typically all approved. Moreover, we do not consider past locations and system notifications to have an impact on future rewards.

To improve the performance of the policy learner and the efficiency of the policy evaluation, we can enhance the Q-function input by incorporating additional features from the full available history. We should only include features that we believe, when combined with the action, are predictive of future rewards. In this context, we include the following features: the time elapsed since the previous approved work order, the cumulative work order costs associated with the box and the refrigeration unit (including emergency repairs), and the cumulative count of live and non-live trips.

The Q-functions are fitted recursively based on the estimated policy for the later stages. For modeling the Q-functions, we utilize a super learner ensemble that combines linear models with different interaction terms and gradient boosting tree models. The entire procedure is cross-fitted using the same 20 folds as for cross-fitting the g-function.

5.4. Policy learning & evaluation

We follow the exact same methodology for policy learning as outlined in the simulation study in Section 4. Both a doubly robust blip learner and a regular Q learner is recursively fitted using the `policy_learn` function from the `polle` R package [Nordland and Holst, 2022]. The blip functions are recursively fitted using a super learner model. The ensemble consists of linear models with different specifications, including additional interaction terms and spline basis expansions of the continuous variables. The variable input for the blip functions is the same as that for the Q-functions.

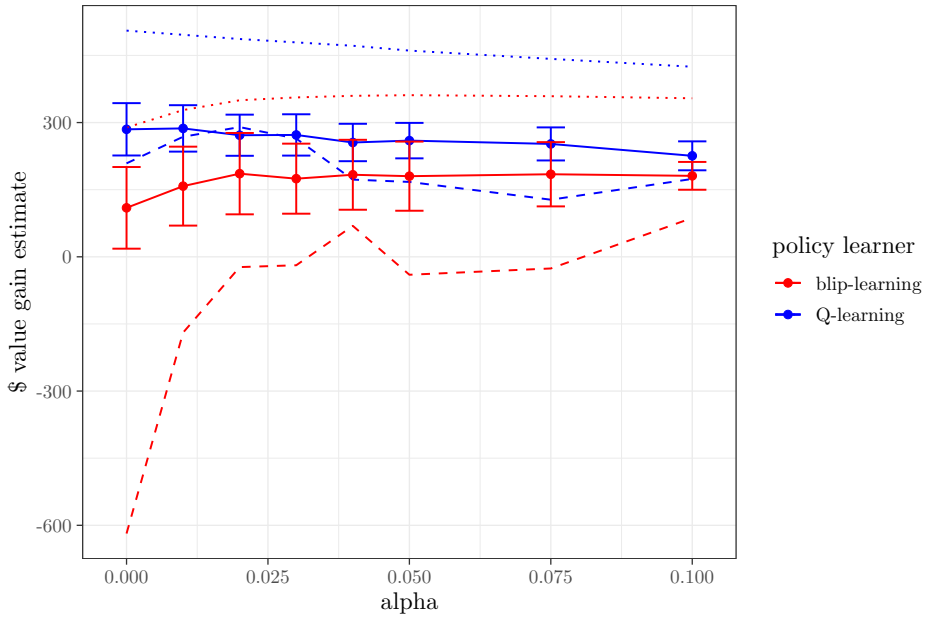


Fig. 9. Cross-fitted value estimates for the policy learners.

Both policy learners are evaluated based on 20-fold cross-fitting, i.e., the g-function, Q-functions, and blip functions are fitted on each of the training folds, and the doubly robust value score is calculated on each of the validation folds. It is important to note that the folds remain the same for every value of the protection level α. The results of the policy evaluation are displayed in Figure 9.

Overall, the Q-learner shows the highest estimated value gain based on the doubly robust value estimator. Specifically, for $\alpha = 0.01$, the estimated value gain is \$287 with a standard error of \$26. For $\alpha = 0.02$ and $\alpha = 0.03$, the estimated gain drops slightly to \$272 with a standard error of \$23. The outcome regression and inverse probability weighting value estimates show some discrepancy, which may indicate issues with consistency of either or both the nuisance models. Without positivity protection, the outcome regression value estimate suggests an optimistic value gain of \$505.

As anticipated, the blip learner is far more sensitive to the positivity protection level compared to the Q-learner. For $\alpha = 0.02$, the estimated value gain is \$186 with a standard error of \$45. Once again, we observe a noticeable discrepancy between the outcome regression estimate and the inverse probability weighting estimate, indicating potential inconsistencies in either or both of the estimates. Based on these results, it is challenging to argue for anything other than the superior performance of the Q-learner, making it the recommended candidate for future implementation.

At this point, our focus is to analyze the implications of the estimated policy derived from the Q-learner. The top figure in Figure 10 represents the number of work orders for which a policy intervention is considered realistic at different α levels. In total, across the 8 stages examined, there are 71,668 observed work orders. However, for a protection level of $\alpha = 0.01$, only 22,313 of these work orders qualify for being controlled by the estimated policy.

The bottom figure in Figure 10 plots the number of work orders for which the observed action is not in agreement with the estimated policy. Interestingly, the total count of disagreements does not vary considerably between $\alpha = 0$ and $\alpha = 0.01$, indicating that the estimated policy without protection does not find value gains in regions of the data with limited support. This suggests that the Q-learner does not heavily rely on extrapolation of the Q-functions as one might have feared.

By further examining the disagreement count between the observed actions and the policy, we see that the policy is more opportunistic compared to the existing action model. Specifically, at $\alpha = 0.01$, the policy recommends approving 4,786 work orders that were in fact rejected, while it only recommends rejecting 1,150 work orders that were in fact approved.

6. Conclusion

Assumption lean statistical policy learning has immense potential within the industry, particularly in the field of physical asset management. It forces the analyst to justify the minimal set of assumption needed to answer causal question from historical data. The transparency of the existing decision process is crucial for justifying sequential randomization, as it requires thorough documentation and storage of all relevant information related to the historical decision-making.

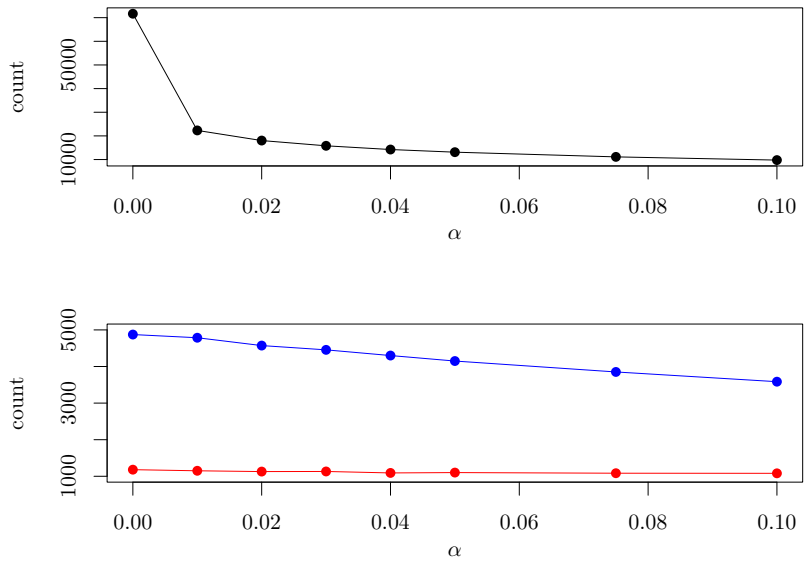


Fig. 10. The top plot presents the count of realistic work orders based on different values of the protection level α. The bottom plot shows the number of observed actions that are not in agreement with the estimated policy. The count is divided into two groups: the blue points represent the count for which the policy recommends approving the work order, while the red points represent the count for which the policy recommend rejecting the work order.

Near positivity violations are a practical challenge that has not received sufficient attention. Even large volumes of historical data have limited value for data-driven decision-making if strict guidelines and policies are in place. Our simulation study and application highlighted the importance of actively restricting a policy learner to consider only realistic alternative actions. This is especially critical for doubly robust policy learners that rely on inverse probability weights. While methods like Q-learning may offer greater stability, they are less robust against model misspecifications. Regardless of the type of realistic policy learner used, we strongly recommend the use of (cross-fitted) doubly robust policy evaluation.

In our specific application, we estimated a significant value gain of \$287 over the lifetime of a refrigerated container. Although this may seem insignificant at first, considering the scale of 300,000 reefers, the potential value gain for the business amounts to \$86 million.

The methods employed in this paper offer a simple starting point for data-driven policy optimization without the requirement to construct more or less realistic simulation models for the system. Any implementation of a learned policy should introduce some stochasticity to the policy for the sake of future optimization. The value of the resulting stochastic policy can be estimated using the same policy evaluation procedure outlined in this paper. Nevertheless, we see possibilities for future research in estimating stochastic policies from historical data under practical positivity violations.

References

Charalampos P Andriotis and Konstantinos G Papakonstantinou. Deep reinforcement learning driven inspection and maintenance planning under incomplete information and constraints. *Reliability Engineering & System Safety*, 212:107551, 2021.

Susan Athey and Stefan Wager. Policy learning with observational data. *Econometrica*, 2020.

Stephane RA Barde, Soumaya Yacout, and Hayong Shin. Optimal preventive maintenance policy based on reinforcement learning of a fleet of military trucks. *Journal of Intelligent Manufacturing*, 30(1):147–161, 2019.

Keith Battocchi, Eleanor Dillon, Maggie Hei, Greg Lewis, Paul Oka, Miruna Oprescu, and Vasilis Syrgkanis. EconML: A Python Package for ML-Based Heterogeneous Treatment Effects Estimation. https://github.com/microsoft/EconML, 2019. Version 0.14.0.

Oliver Bembom and Mark J van der Laan. A practical illustration of the importance of realistic individualized treatment rules in causal inference. *Electronic journal of statistics*, 1:574, 2007.

Bibhas Chakraborty and EE Moodie. *Statistical methods for dynamic treatment regimes*. Springer, 2013.

Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, Mu Li, Junyuan Xie, Min Lin, Yifeng Geng, Yutian Li, and Jiaming Yuan. *xgboost: Extreme Gradient Boosting*, 2023. URL `https://CRAN.R-project.org/package=xgboost`. R package version 1.7.3.1.

Yuan Chen, Ying Liu, Donglin Zeng, and Yuanjia Wang. *DTRlearn2: Statistical Learning Methods for Optimizing Dynamic Treatment Regimes*, 2020. URL `https://CRAN.R-project.org/package=DTRlearn2`. R package version 1.1.

Stephen R Cole and Miguel A Hernán. Constructing inverse probability weights for marginal structural models. *American journal of epidemiology*, 168(6):656–664, 2008.

Bram De Jonge and Philip A Scarf. A review on maintenance optimization. *European journal of operational research*, 285 (3):805–824, 2020.

Alexander D'Amour, Peng Ding, Avi Feller, Lihua Lei, and Jasjeet Sekhon. Overlap in observational studies with high-dimensional covariates. *Journal of Econometrics*, 221(2):644–654, 2021.

José Manuel Torres Farinha. *Asset maintenance engineering methodologies*. CRC Press, 2018.

Yair Goldberg and Michael R Kosorok. Q-learning with censored data. *Annals of statistics*, 40(1):529, 2012.

Niclas Hoffmann, Robert Stahlbock, and Stefan Voß. A decision model on the repair and maintenance of shipping containers. *Journal of Shipping and Trade*, 5(1):1–21, 2020.

S. T. Holloway, E. B. Laber, K. A. Linn, B. Zhang, M. Davidian, and A. A. Tsiatis. *DynTxRegime: Methods for Estimating Optimal Dynamic Treatment Regimes*, 2022. URL `https://CRAN.R-project.org/package=DynTxRegime`. R package version 4.11.

Edward H Kennedy. Towards optimal doubly robust estimation of heterogeneous causal effects. *arXiv preprint arXiv:2004.14497*, 2020.

Michael R Kosorok and Eric B Laber. Precision medicine. *Annual review of statistics and its application*, 6:263–286, 2019.

Yu Liu, Yiming Chen, and Tao Jiang. Dynamic selective maintenance optimization for multi-state systems over a finite horizon: A deep reinforcement learning approach. *European Journal of Operational Research*, 283(1):166–181, 2020.

Alex Luedtke and Incheoul Chung. One-step estimation of differentiable hilbert-valued parameters. *arXiv preprint arXiv:2303.16711*, 2023.

Alex Luedtke, Antoine Chambaz, et al. Performance guarantees for policy learning. In *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, volume 56, pages 2162–2188. Institut Henri Poincaré, 2020.

Alexander R Luedtke and Mark J van der Laan. Super-learning of an optimal dynamic treatment rule. *The international journal of biostatistics*, 12(1):305–332, 2016.

Lina M Montoya, Mark J van der Laan, Alexander R Luedtke, Jennifer L Skeem, Jeremy R Coyle, and Maya L Petersen. The optimal dynamic treatment rule superlearner: considerations, performance, and application to criminal justice interventions. *The International Journal of Biostatistics*, 2022.

Kelly L Moore, Romain Neugebauer, Mark J van der Laan, and Ira B Tager. Causal inference in epidemiological studies with strong confounding. *Statistics in medicine*, 31(13):1380–1404, 2012.

Susan A Murphy. Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(2):331–355, 2003.

Xinkun Nie, Emma Brunskill, and Stefan Wager. Learning When-to-Treat Policies. *Journal of the American Statistical Association*, 116(533):392–409, 2021. doi: 10.1080/01621459.2020.1831925.

Andreas Nordland and Klaus K. Holst. *polle: Policy Learning*, 2022. URL `https://CRAN.R-project.org/package=polle`. R package version 1.3.

Andreas Nordland and Klaus K. Holst. Policy learning with the polle package, 2023.

Maya L Petersen, Kristin E Porter, Susan Gruber, Yue Wang, and Mark J van der Laan. Diagnosing and responding to violations in the positivity assumption. *Statistical methods in medical research*, 21(1):31–54, 2012.

Eric Polley, Erin LeDell, Chris Kennedy, and Mark van der Laan. *SuperLearner: Super Learner Prediction*, 2021. URL `https://CRAN.R-project.org/package=SuperLearner`. R package version 2.0-28.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

James Robins. A new approach to causal inference in mortality studies with a sustained exposure period—application to control of the healthy worker survivor effect. *Mathematical modelling*, 7(9-12):1393–1512, 1986.

James M Robins. Optimal structural nested models for optimal sequential decisions. In *Proceedings of the second seattle Symposium in Biostatistics*, pages 189–326. Springer, 2004.

Thomas H. Scheike, Klaus K. Holst, and Jacob B. Hjelmborg. Estimating heritability for cause specific mortality based on twin studies. *Lifetime Data Analysis*, 20(2):210–233, 2014. doi: 10.1007/s10985-013-9244-x.

Phillip J. Schulte, Anastasios A. Tsiatis, Eric B. Laber, and Marie Davidian. Q- and a-learning methods for estimating optimal dynamic treatment regimes. *Statistical Science*, 29(4):640–661, 2014. doi: 10.1214/13-STS450.

Vira Semenova and Victor Chernozhukov. Debiased machine learning of conditional average treatment effects and other causal functions. *The Econometrics Journal*, 24(2):264–289, 2021.

Rengarajan Srinivasan and Ajith Kumar Parlikad. Semi-markov decision process with partial information for maintenance decisions. *IEEE Transactions on Reliability*, 63(4):891–898, 2014.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Anastasios A Tsiatis. *Dynamic Treatment Regimes: Statistical Methods for Precision Medicine*. CRC Press, 2019.

Masatoshi Uehara and Wen Sun. Pessimistic model-based offline reinforcement learning under partial coverage, 2023.

S.N Wood. *Generalized Additive Models: An Introduction with R*. Chapman and Hall/CRC, 2 edition, 2017.

Nailong Zhang and Wujun Si. Deep reinforcement learning for condition-based maintenance planning of multi-component systems under dependent competing risks. *Reliability Engineering & System Safety*, 203:107094, 2020.

Yingqi Zhao, Donglin Zeng, A John Rush, and Michael R Kosorok. Estimating individualized treatment rules using outcome weighted learning. *Journal of the American Statistical Association*, 107(499):1106–1118, 2012.

Paper C

Estimation of treatment effect among treatment responders with a time-to-event endpoint

Authors:
Andreas Nordland & Torben Martinussen

Publication details:
Accepted for publication in the Scandinavian Journal of Statistics

ORIGINAL ARTICLE

Estimation of treatment effect among treatment responders with a time-to-event endpoint

Andreas Nordland[1*] | **Torben Martinussen**[1]

[1]Section of Biostatistics, University of Copenhagen

Correspondence
Andreas Nordland, Section of Biostatistics, University of Copenhagen, Øster Farimagsgade 5, DK-1014 Copenhagen, Denmark
Email: anno@sund.ku.dk

Funding information
Innovation Fund Denmark, grant number 8053-00096B

In a placebo-controlled clinical study one may calculate the average treatment effect to convey the effect of the active treatment on some outcome. However, if it is speculated that the treatment only has an effect if the patient responds to the treatment defined by a certain biomarker response, then it is arguably more relevant to estimate the treatment effect among such responders. We present such a causal parameter that is based on principal stratification and is identified under the exclusion of a treatment effect among the non-responders. We focus on time-to-event outcomes allowing for right censoring, and construct a doubly robust and efficient estimator based on the associated efficient influence function. The properties of the estimator are showcased in a simula-

tion study and the methodology is applied to the Leader trial investigating the effect of liraglutide on the occurrence of cardiovascular events.

KEYWORDS

causal inference, compliance, efficient influence function, local average treatment effect, principal stratification, survival, treatment effect among responders

1 | INTRODUCTION

In a placebo controlled clinical trial the average treatment effect is clearly an estimand of great interest. However, in situations with a potential long term outcome such as a time-to-event it may also be of interest to compare only populations where it is expected that the active treatment should work. Such insight may sometimes be obtained via biomarker measurements as described for instance in Bornkamp and Bermann (2019) (BB). Specifically, they describe the CANTOS outcomes study (Ridker et al., 2017) where the primary outcome was time to a major adverse cardiac event (MACE). The CANTOS trial investigated treatment with an anti-inflammatory agent against placebo as inflammation has been identified as a important factor in atherosclerosis. In that study, the used biomarker was high sensitivity C-reactive protein (hs-CRP) with lower values indicating less inflammation. Interest focused on the treatment effect on MACE for patients that would have their hs-CRP lowered beyond a specific target level three months after treatment initiation. Because, otherwise it is suspected that treatment will have no effect on the long term outcome. Such a comparison is, however, not straight forward as also detailed in BB. A fair comparison is as mentioned to focus on the patients that would have the desired biomarker response if treated with the active agent, but this biomarker response is unknown for patients in the placebo arm. We note that the considered problem is mirrored by the problem of estimating the causal treatment effect among those that comply with the active treatment in a placebo-controlled clinical study where there may be patients that do not comply with the active treatment. Again, the compliance status is not observed for patients in the placebo arm. The population of interest here is also known as a principal stratum and methods for dealing with such problems has attracted some interest lately as nicely summarized in Bornkamp et al. (2021) that also give further examples where a principal strata estimand may be of clinical

interest. We assume a stochastic exclusion criteria to identify the causal estimand of interest but other identification assumptions exist as well, see Stuart and Jo (2015); Larsen and Josiassen (2020); Jiang and Ding (2021). In this paper we focus on the situation with a time-to-event outcome that may be right censored, and we wish to estimate the before mentioned causal effect among treatment responders just like the situation in the CANTOS study. Such an estimation procedure was also proposed by BB. However, the method we propose improves upon their estimator in several important ways. The plug-in estimator proposed by BB requires completely independent censoring and correct specification of the proposed working Cox-model. Both assumptions are likely to fail in many studies leading to biased estimation of the target estimand. We further allow for the use of baseline covariates like BB, but develop estimation based on the corresponding efficient influence function the benefit being a robust and efficient estimation procedure that only requires conditional independent censoring given the covariates. Furthermore, the covariates are used in an active way to gain precision, which is possible due to the initial randomization of treatment. An important feature of our method is also that censoring is allowed to depend on the biomarker response as non-responders may be more likely to drop-out of the study. To our knowledge there is no such estimator available for this specific estimand in the literature. We furthermore derive the asymptotic properties of the estimator making formal inference possible, and investigate the estimators performance in simulation studies. We also apply the presented methods to data from the LEADER trial (Marso et al., 2016). The trial investigates the effect of liraglutide on cardiovascular events. In Section 2, inspired by the well-known local average treatment effect (Angrist et al., 1996; Frangakis and Rubin, 2002), we formally define responders as a principal stratum resulting in a causally interpretable average treatment effect for that stratum. We identify the causal parameter by the (stochastic) exclusion of a treatment effect among the non-responders and we discuss situations where the restriction is a reasonable assumption. In Section 3 we present the efficient estimator of the resulting target parameter based on the associated efficient influence functions taking into account the baseline covariates and that we have a randomized study design (Bickel et al., 1993; Van der Vaart, 2000; Van der Laan et al., 2003; Hines et al., 2022). For generality, we start handling the situation with binary and continuous outcomes and then turn to the before mentioned time-to-event outcome with censoring. Section 4 demonstrates the properties of the constructed estimator in two simulation studies. Finally, in Section 5 we apply the presented methods based on data from the LEADER trial (Marso et al., 2016). Our analysis joins an increasing list of clinical applications (Bornkamp and Bermann, 2019; Larsen and Josiassen, 2020; Bornkamp et al., 2021; Hirano et al., 2000; Loeys and Goetghebeur, 2003; Gilbert et al., 2003; Egleston et al., 2017; Magnusson et al., 2019).

2 | SETUP AND THE AVERAGE TREATMENT EFFECT AMONG RESPONDERS

2.1 | Continuous or Binary Outcome

Let $A \in \{0,1\}$ denote a randomization indicator in a clinical trial with $A = 1$ corresponding to treatment and $A = 0$ to placebo. The randomization probability $\mathbb{P}(A = 1) = \delta \in (0,1)$ is assumed to be known. Let $D \in \{0,1\}$ denote a post-randomization indicator (e.g. an indicator function of biomarkers). Let $W \in \mathcal{W}$ denote a vector of baseline variables. The outcome of interest is denoted $Y \in \mathcal{Y}$ and may be continuous or binary. Later we focus on the situation with a time-to-event outcome, which may be right censored. We assume that observations are generated from the structural model

$$W = f_W(U_W), \quad A = f_A(U_A),$$
$$D = f_D(W, A, U_D), \quad Y = f_Y(W, A, D, U_Y),$$

where $U = (U_W, U_A, U_D, U_Y)$ is a vector of exogenous variables. Due to randomization, U_A is independent of the other exogenous variables. Let $O = (W, A, D, Y)$ denote the observed data and let P_0 denote the true distribution of the observed data. The data generating model allow us to formulate potential outcomes as a result of an intervention on the assignment of treatment

$$D(a) = f_D(W, a, U_D), \quad Y(a) = f_Y(W, a, D(a), U_Y).$$

Importantly, both $D(1)$ and $D(0)$ are independent of A and thus behave as proper baseline variables. On this basis, Angrist et al. (1996) and Imbens and Rubin (1997) defines four principal strata denoted *compliers* $\{D(0) = 0, D(1) = 1\}$, *always-takers* $\{D(0) = 1, D(1) = 1\}$, *defiers* $\{D(0) = 1, D(1) = 0\}$ and *never-takers* $\{D(0) = 0, D(1) = 0\}$. Note that the principal stratum of an individual can never be observed. However, the stratum of *responders* $\{D(1) = 1\}$ composed of compliers and always-takers is observed for individuals randomized to treatment but not for individuals randomized to placebo. We define *the average treatment effect among responders* as

$$\mathbb{E}[Y(1) - Y(0) | D(1) = 1]. \tag{1}$$

To identify the average treatment effect among responders, we rely on the *stochastic exclusion restriction*, see Hirano et al. (2000), defined as

$$\mathbb{E}[Y(1) - Y(0)|D(1) = 0] = 0. \tag{2}$$

Under the stochastic exclusion restriction,

$$\mathbb{E}[Y(1) - Y(0)] = \mathbb{E}[Y(1) - Y(0)|D(1) = 1]\mathbb{P}[D(1) = 1],$$

and thus the average treatment effect among responders is identified as

$$\begin{aligned}\mathbb{E}[Y(1) - Y(0)|D(1) = 1] &= \frac{\mathbb{E}[Y(1) - Y(0)]}{\mathbb{P}[D(1) = 1]} \\ &= \frac{\mathbb{E}[Y|A = 1] - \mathbb{E}[Y|A = 0]}{\mathbb{E}[D|A = 1]},\end{aligned} \tag{3}$$

where the last equality holds due to consistency. This result is similar to Proposition 1 in Angrist et al. (1996). Note that the estimand has some resemblance to the Wald estimand in the instrumental variables setting, see Didelez and Sheehan (2007). For ease of notation let

$$\begin{aligned}\Psi^a(P) &= \mathbb{E}[Y|A = a] = \mathbb{E}\{\mathbb{E}[Y|A = a, W]\}, \\ \Psi^D(P) &= \mathbb{E}[D|A = 1] = \mathbb{E}\{\mathbb{E}[D|A = 1, W]\}.\end{aligned}$$

That $\mathbb{E}[Y|A = a] = \mathbb{E}\{\mathbb{E}[Y|A = a, W]\}$ follows because of A and W being independent, and similarly with the last equation in the latter display.

2.2 | Time to Event Outcome

Let A, D and W be defined as before. In this setup the outcome of interest is a time-to-event endpoint T, which may be subject to right censoring. Let C denote the censoring time. The observed outcome is given by $\tilde{T} = \min(T, C)$ and $\Delta = I\{T < C\}$. We assume that $T \perp C|W, A, D$. It is important that the censoring is allowed to depend both on (A, W) but also on D as this is likely to be the case in many practical settings where censoring may be more likely for patients where the biomarker information indicates treatment failure. Let $O = (W, A, D, \tilde{T}, \Delta)$ denote the observed

data. As before, we assume that the outcome is generated from a structural model

$$T = f_T(W, A, D, U_T), \quad C = f_C(W, A, D, U_C),$$

where U_T and U_C are exogenous variables related to the outcome. Note that $U_T \perp U_C | W, A, D$. Let P_0 denote the true distribution of the observed data O with density

$$p_0(o) = \mu_0(w) g_0(a) h_0(d|w,a) \lambda_0(t|w,a,d)^{\Delta} S_0(t|w,a,d) \lambda_0^c(t|w,a,d)^{(1-\Delta)} S_0^c(t|w,a,d),$$

where λ_0 and λ_0^c denote the hazard functions of T and C and S_0 and S_0^c denote the survival functions of T and C. Again, the data generating model allow us to formulate potential outcomes as a result of an intervention on the assignment of treatment

$$T(a) = f_T(W, a, D(a), U_T), \quad C(a) = f_C(W, a, D(a), U_C).$$

For a given time point $\tau > 0$ we are interested in the average treatment effect among responders defined as

$$\mathbb{P}(T(1) \leq \tau | D(1) = 1) - \mathbb{P}(T(0) \leq \tau | D(1) = 1). \tag{4}$$

For identification purposes, we assume positivity for the censoring variable, i.e., we assume that $S_0^c(\tau|W, A, D) > \eta$ almost surely for some $\eta > 0$. Under the stochastic exclusion restriction (2), the average treatment effect among responders can be identified similar to (3) as

$$\frac{\mathbb{P}(T \leq \tau | A = 1) - \mathbb{P}(T \leq \tau | A = 0)}{\mathbb{E}[D|A = 1]}, \tag{5}$$

where

$$\mathbb{P}(T \leq \tau | A = a) = \mathbb{E}\{\mathbb{P}(T \leq \tau | A = a, W)\}$$

because of the randomization. For ease of notation let $\Psi^a(P) = \mathbb{P}_P(T \leq \tau | A = a)$.

2.3 | Notes on the Stochastic Exclusion Restriction

The stochastic exclusion restriction (2) formulated in this paper states that the average treatment effect for non-responders is zero, i.e., that the treatment has no effect on the outcome if the respondence under treatment is zero. This formulation is weaker than the *monotonicity assumption* and exclusion restriction originally stated in Angrist et al. (1996). Here, the authors assume that

$$D(1) \geq D(0) \tag{6}$$

$$Y(A = a, D = d) = Y(A = a', D = d) \quad \forall a, a', d \in \{0, 1\}. \tag{7}$$

Due to monotonicity (6), on the set $\{D(1) = 0\}$ it must also hold that $D(0) = 0$, and due to the exclusion restriction (7) $Y(1, 0) = Y(0, 0)$. Thus, it holds that

$$\begin{aligned}
\mathbb{E}[Y(1) - Y(0)|D(1) = 0] &= \mathbb{E}[Y(1, D(1)) - Y(0, D(0))|D(1) = 0, D(0) = 0] \\
&= \mathbb{E}[Y(1, 0) - Y(0, 0)|D(1) = 0, D(0) = 0] \\
&= 0.
\end{aligned}$$

It is clear that the exclusion restriction (7) can be replaced by the restriction that

$$Y(A = 1, D = 0) = Y(A = 0, D = 0). \tag{8}$$

The disadvantage of (8) and (7) is that these restrictions only make sense in situations where an intervention on D is hypothetically possible. The stochastic exclusion restriction (2) does not suffer from the same conceptual issue.

The stochastic exclusion restriction is nevertheless an untestable assumption, so whether it is reasonable depends on expert knowledge about the specific application. An ideal application is a blinded randomized trial with all-or-nothing compliance where patients assigned to placebo do not have access to the active treatment, e.g., the patient has to take a single pill. In this case, the post-randomization indicator expresses the exposure to the active treatment. Because the trial is blinded, it is reasonable to assume that the treatment effect is zero for non-responders.

Another important case is a blinded randomized trial where post-randomization biomarkers identify all or a subset of patients not reacting to the active treatment. If a medical expert can argue that the treatment itself cannot have an effect on the outcome of interest for this group of patients, then it

is again reasonable to assume that the stochastic exclusion restriction holds.

3 | EFFICIENT ESTIMATION

The target parameter for estimation has the form

$$\Psi^T(P_0) = \frac{\Psi^1(P_0) - \Psi^0(P_0)}{\Psi^D(P_0)}. \tag{9}$$

We aim to efficiently estimate each of the subparameters $\Psi^1(P_0)$, $\Psi^0(P_0)$, and $\Psi^D(P_0)$ and then efficiently estimate $\Psi^T(P_0)$ by plug-in. The key to constructing efficient and asymptotically linear estimators for each of the subparameters is to derive the corresponding *efficient influence functions* (Bickel et al., 1993; Van der Vaart, 2000; Van der Laan et al., 2003), see the Appendix for a brief introduction to this theory.

3.1 | Continuous or Binary Outcome

Since the orthogonal complement of the tangent space is given by $\mathcal{T}^{\perp} = \left\{(A-\delta)h_*(W) : \mathbb{E}\left[h_*(W)^2\right] < \infty\right\}$ it can be shown that the efficient influence functions for $\Psi^D(P_0) = \mathbb{E}[D|A=1]$ and $\Psi^a(P_0) = \mathbb{E}[Y|A=a]$ are given by

$$\tilde{\psi}^D(P_0)(O) = \frac{A}{g_0(1)}\left(D - H_0(W)\right) + H_0(W) - \Psi^D(P_0), \tag{10}$$

$$\tilde{\psi}^a(P_0)(O) = \frac{I\{A=a\}}{g_0(a)}\left(Y - Q_0(a,W)\right) + Q_0(a,W) - \Psi^a(P_0), \tag{11}$$

where

$$g_0(a) = \mathbb{P}(A=a) = a\delta + (1-a)(1-\delta),$$

$$H_0(w) = \mathbb{E}[D|A=1, W=w],$$

$$Q_0(a,w) = \mathbb{E}[Y|A=a, W=w],$$

see for example Tsiatis (2006). Let $\hat{H}_n$ and $\hat{Q}_n$ denote models for H_0 and Q_0 and define the one-step estimators

$$\tilde{\Psi}_n^D(\hat{H}_n) = P_n\tilde{\phi}^D(\hat{P}_n)(O) = P_n\left\{\frac{A}{g_0(1)}\left(D - \hat{H}_n(W)\right) + \hat{H}_n(W)\right\} \tag{12}$$

$$\tilde{\Psi}_n^a(\hat{Q}_n) = P_n\tilde{\phi}^a(\hat{P}_n)(O) = P_n\left\{\frac{I(A=a)}{g_0(a)}\left(Y - \hat{Q}_n(a,W)\right) + \hat{Q}_n(a,W)\right\}. \tag{13}$$

We assume that

$$\|\hat{H}_n - H^*\|_{P_0} = o_{P_0}(1), \quad \|\hat{Q}_n - Q^*\|_{P_0} = o_{P_0}(1), \tag{14}$$

for some models H^* and Q^*. As described in Section 3, to prove that the one-step estimators $\tilde{\Psi}_n^D(\hat{H}_n)$ and $\tilde{\Psi}_n^a(\hat{Q}_n)$ have influence functions $\tilde{\psi}^D(P^*)(O)$ and $\tilde{\psi}^a(P^*)(O)$, we need to prove that the associated remainder terms are $o_{P_0}(n^{-1/2})$. It is simple to show that

$$P_0\tilde{\phi}^D(\hat{P}_n)(O) = \Psi^D(P_0)$$
$$P_0\tilde{\phi}^a(\hat{P}_n)(O) = \Psi^a(P_0), \ a \in \{0,1\}.$$

Thus, the second order remainders $\tilde{R}^D(\hat{P}_n, P_0)$ and $\tilde{R}^a(\hat{P}_n, P_0)$ are zero, see (20) in the Appendix. We are left to prove that the empirical process remainder, given by (21) in the Appendix, is $o_{P_0}(n^{-1/2})$. In particular, this will be the case if the models $\hat{H}_n$ and $\hat{Q}_n$ fall in a Donsker class with probability tending to one. The Donsker class condition holds for parametric models, but it will generally not hold for data-adaptive models for which the complexity increase with the sample size, see Chernozhukov et al. (2018). However, the Donsker class condition can be avoided by fitting $\hat{H}_n$ and $\hat{Q}_n$ on a separate data set, for example, by constructing a cross-fitted one-step estimator as described by Chernozhukov et al. (2018). Note that (14) is a very weak model requirement. Thus, we are almost guaranteed that the one-step estimators (12) and (13) are consistent and that inference based on an approximation of (23) has the correct level of coverage. As a result, the associated plug-in estimator of the average treatment effect among responders given by (24) will also be consistent and have the correct level of coverage. Lastly, the plug-in estimator will be asymptotically efficient if $H^* = H_0$ and $Q^* = Q_0$.

3.2 | Time to Event Outcome

The efficient influence function for $\Psi^D(P_0) = \mathbb{E}[D|A=1]$ is still given by equation (10) with an associated one-step estimator given by (12). Thus, we can focus on estimating the subparameter $\Psi^a(P_0) = \mathbb{P}(T \leq \tau | A = a)$. For ease of notation, let $X = (W, A, D)$. As shown in the Appendix, the efficient influence function for $\Psi^a(P_0)$ is given by

$$\begin{aligned}\tilde{\psi}^a(P_0)(O) &= \frac{I\{A=a\}}{g_0(a)}\left(\frac{\Delta(\tau)}{S_0^c(\min(\tilde{T},\tau)|X)} I\{\tilde{T} \leq \tau\} + \int_0^\tau \frac{S_0(u|X) - S_0(\tau|X)}{S_0(u|X)S_0^c(u|X)} dM_0^c(u|X))\right) \\ &+ \left(1 - \frac{I\{A=a\}}{g_0(a)}\right) F_0(\tau|A=a,W) - \Psi^a(P_0) \qquad (15) \\ &= \tilde{\phi}^a(P_0) - \Psi^a(P_0),\end{aligned}$$

where $\Delta(\tau) = I\{C > \min(T,\tau)\}$ and

$$\begin{aligned}F_0(\tau|A=a,W) &= \sum_{d\in\{0,1\}} \left\{\int_0^\tau S_0(u|W,a,d) d\Lambda_0(u|W,a,d)\right\} h_0(d|W,a), \qquad (16) \\ M_0^c(s|X) &= I\{\tilde{T} \leq s, \Delta = 0\} - \int_0^s I\{\tilde{T} \geq u\} d\Lambda_0^c(u|X),\end{aligned}$$

the latter being the censoring martingale. Define the one-step estimator

$$\tilde{\Psi}_n^a(\hat{P}_n) = P_n \tilde{\phi}^a(\hat{P}_n)(O), \qquad (17)$$

for a model $\hat{P}_n$ of $\hat{\Lambda}_n$, $\hat{\Lambda}_n^c$ and $\hat{h}_n$ with limiting model P^*. Related work is given in Lee et al. (2023), where an instrumental variable setting is considered in a discrete time setup. To show that the one-step estimator has influence function $\tilde{\phi}^a(P^*)(O) - \mathbb{E}[\tilde{\phi}^a(P^*)(O)]$, we need to prove that second order remainder (20) and empirical process remainder (21) are $o_{P_0}(n^{-1/2})$. General model conditions ensuring these properties are still under active research, see Westling et al. (2021) and Rytgaard et al. (2021). As shown in the appendix, the second order remainder is given by

$$\begin{aligned}R^a(\hat{P}_n, P_0) = P_0\Bigg[&- \frac{I\{A=a\}}{g_0(a)} \int_0^\tau \left\{\frac{\hat{S}_n(\tau|X)}{\hat{S}_n(u|X)}\left[S_0(u|X) - \hat{S}_n(u|X)\right] - \left[S_0(\tau|X) - \hat{S}_n(\tau|X)\right]\right\} \\ &\times \frac{S_0^c(u|X)}{\hat{S}_n^c(u|X)}\left\{d\Lambda_0^c(u|X) - d\hat{\Lambda}_n^c(u|X)\right\}\Bigg].\end{aligned}$$

The remainder is well defined if we assume that positivity holds for the censoring model as well, i.e., that $\hat{S}_n^c(\tau|X) > \eta$ for some $\eta > 0$. As noted by Westling et al. (2021), the second order remainder and empirical process remainder are $o_{P_0}(n^{-1/2})$ if both the time-to-event and censoring distributions are correctly modeled by a Cox proportional hazard model. However, the product structure of the second order remainder indicates that more flexible modelling of the time-to-event and censoring distributions are possible. Furthermore, we note that $R^a(P^*, P_0) = 0$ if either $\Lambda^* = \Lambda_0$ or $\Lambda^{c*} = \Lambda_0^c$, meaning that the one-step estimator is consistent if either the time-to-event or censoring distribution is correctly modelled. The one-step estimator will be efficient if both $\Lambda^* = \Lambda_0$ and $\Lambda^{c*} = \Lambda_0^c$.

4 | SIMULATIONS

4.1 | Binary Outcome

The purpose of this simulation example is to demonstrate the guaranteed consistency of the efficient one-step estimator described in Section 3.1, as well as to illustrate the gain in efficiency over the usual empirical mean plug-in estimate. The R code is publicly available at `https://github.com/Andreas-Nordland/rate`. We let $\mathbb{P}(A = 1) = \delta = 0.5$ and simulate the same number of observations in the treatment and placebo groups. We sample from a structural model given by

$$
\begin{aligned}
W_k &\sim \mathcal{N}(0,1), \qquad k \in \{1,\dots,10\}. \\
D|W_1,W_2,W_3,A &\sim \begin{cases} \text{Ber}(\text{expit}(2I\{W_1 > 0\}\sin(2W_2) + \exp(W_3)) & A = 1 \\ 0 & A = 0 \end{cases}, \\
Y|D,W_1,W_4,W_5 &\sim \text{Ber}\left(\text{expit}(D(2\cos(2W_4) - 1) + W_1W_5 + \log(|W_5W_4|))\right).
\end{aligned}
$$

The efficient plug-in estimate of the average treatment effect among responders relies on nuisance models for $H_0(W) = \mathbb{E}(D|A = 1, W)$ and $Q_0(a, W) = \mathbb{E}[Y|A = a, W]$ for $a \in \{0, 1\}$. We fit both models using an ensemble of models, where each model receives a weight based on the cross-validated loss of the fit. Specifically, we apply the super learner method described by Van der Laan et al. (2007). The ensembles include logistic regression, generalized additive model regression with integrated smoothness estimation, and random forest regression. The estimation procedure is cross-fitted to relax the restrictions on the model complexity, see Chernozhukov et al. (2018).

The results of the simulation are given in Table 1. As anticipated, the efficient plug-in estimate is

consistent and shows the correct level of coverage even though both nuisance models are based on flexible regression models. We see a clear improvement in performance with a standard deviation ratio of 0.92 when compared to the empirical mean plug-in estimate.

Table 1 about here

4.2 | Time-to-Event Outcome

This simulation example is intended to demonstrate the importance of conditioning on the response indicator for ensuring conditionally independent censoring. It also illustrates that incorporating predictive baseline variables can enhance efficiency. The detailed description of the efficient one-step estimator can be found in Section 3.2. The corresponding R code is openly accessible at `https://github.com/Andreas-Nordland/rate`. In this simulation, we set $\mathbb{P}(A = 1) = \delta = 0.5$ and generate an equal number of observations for both treatment and placebo groups. The data is sampled from a structural model defined as follows:

$$
\begin{aligned}
W &\sim \text{Unif}(0,3), \\
D|W,A &\sim \begin{cases} \text{Ber}(\text{expit}(\kappa^T[1,W])) & A = 1 \\ 0 & A = 0 \end{cases}, && \kappa = [2,-0.5], \\
T|W,D &\sim \lambda_0(t|W,D) = \exp\left(\beta^T[1,W,D,D\cdot W]\right), && \beta = [-2,2,-0.2,-0.4], \\
C|A,D &\sim \lambda_0^c(t|A,D) = \exp\left(\zeta^T[1,A,D]\right), && \zeta = [1,1,-1],
\end{aligned}
$$

where λ_0 and λ_0^c are the hazard functions. We are interested in the average treatment effect among responders at time point $\tau = 0.5$. It is easy to see that the above structural equation model implies both monotonicity and exclusion as described in Section 2.3. Thus, the treatment effect among non-responders is zero and the estimator will have the desired causal interpretation.

The efficient one-step estimate relies on nuisance models for $H_0(W) = \mathbb{E}(D|A = 1, W)$ and the cumulative hazard functions $\Lambda_0(\cdot|X)$ and $\Lambda_0^c(\cdot|X)$ where $X = (W, A, D)$. A model for $H_0(W)$ is fitted using maximum likelihood estimation. The cumulative hazard functions are fitted using Cox proportional hazards models with all interactions.

As evident from the structural equation model, it holds that $T \perp C|A, D$, meaning that the baseline variable W in this scenario solely contributes to enhancing the efficiency of the one-step estimator. Improved efficiency is expected only if W, possibly in combination with the treatment

A, is predictive of the event time. Our model reveals a strong association between W and T. To illustrate the efficiency gain, we also compute the efficient one-step estimate without access to W, i.e., $X = (A, D)$. Finally, we demonstrate that omitting the response indicator D in the cumulative event time and censoring time hazard models leads to bias. Specifically, we use working Cox-models for the two hazard function taking $X = (W, A)$, and we also calculated the simple plug-in estimator based on naive Kaplan-Meier estimates for the two treatment strata. The results of the simulation are presented in Table 2.

Table 2 about here

As anticipated, the estimators based on $X = (W, A, D)$ and $X = (A, D)$ are both consistent and exhibit the correct coverage level. However, the estimator that includes W demonstrates a 10% reduction in standard deviation. On the other hand, the final estimator based on inputs $X = (W, A)$ is evidently biased, leading to an incorrect coverage level. The plug-in estimator using Kaplan-Meier estimates for the two treatment strata is also biased as expected.

5 | APPLICATION

We now apply the proposed methods to the LEADER trial data investigating the effect of liraglutide on cardiovascular events for type 2 diabetics, see Marso et al. (2016). In short, we use the change in glycated hemoglobin to define respondence and find a 9% increase in the average treatment effect among responders.

A total of 9340 patients at least 50 years of age with type 2 diabetes and a high risk for cardiovascular disease were randomized (1:1 ratio) to receive liraglutide or placebo. Liraglutide is an analogue of human glucagon-like peptide 1, which stimulates insulin production and thus decreases blood sugar levels. A high percentage of glycated hemoglobin is indicative of high blood sugar levels. Only patients with a glycated hemoglobin level of 7% or more were eligible to enter the trial. We consider a composite time-to-event endpoint defined as the first occurrence of non-fatal stroke, non-fatal myocardial infarction, and all-cause death. The time-to-event endpoint is measured in months and the minimum planned follow-up was 42 months, with a maximum of 60 months of receiving the assigned regimen. A total of 1572 patients experienced an event within the follow-up period.

The baseline variables considered in this analysis include sex, smoking status (never/prior/current), BMI group (<=30/>30), prior cardiovascular events (yes/no), antidiabetic therapy group, diabetes duration group (<=11 years/>11 years), calculated eGFR-MDRD and the percentage of glycated

hemoglobin (at the time of randomization). 112 patients are excluded from the analysis due to missing baseline variables.

The level of glycated hemoglobin was measured over time to assess the effectiveness of the drug. Let H_b denote the baseline percentage level of glycated hemoglobin and let H_5 denote the percentage level of glycated hemoglobin at visit 5 scheduled 3 months after randomization. We define the post-randomization respondence indicator as

$$D^\beta = 1 - I\{H_b \geq 7.5\} I\{H_5 \geq H_b - \beta\}, \tag{18}$$

for a threshold parameter β. Thus, a non-responder is defined as a patient with elevated glycated hemoglobin levels (above 7.5%) who does not experience a decrease in glycated hemoglobin of at least β. Lower values of β will classify fewer patients as non-responders and as a result, the target estimate will be more conservative.

Obviously, it is possible to experience an event or drop out before visit 5 where the biomarker is measured. Since visit 5 is scheduled 3 months after randomization, it is reasonable to assume that by month 4, if no event or drop out has occurred, it would have been possible to measure the level of glycated hemoglobin. We adjust the principal stratum accordingly and consider the causal target parameter given by:

$$\mathbb{P}(T(1) > 50 | \tilde{T}(1) > 4, D^\beta(1) = 1) - \mathbb{P}(T(0) > 50 | \tilde{T}(1) > 4, D^\beta(1) = 1).$$

Under the stochastic exclusion restriction

$$\mathbb{P}(T(1) > 50 | \tilde{T}(1) > 4, D^\beta(1) = 0) - \mathbb{P}(T(0) > 50 | \tilde{T}(1) > 4, D^\beta(1) = 0) = 0,$$

the causal target parameter equals

$$\frac{\mathbb{P}(T(1) > 50 | \tilde{T}(1) > 4) - \mathbb{P}(T(0) > 50 | \tilde{T}(1) > 4)}{\mathbb{P}(D^\beta(1) = 1 | \tilde{T}(1) > 4)}.$$

If we assume that treatment has no effect up until month 4 on both the event time and the censoring time, i.e., that $I\{\tilde{T}(1) > 4\} = I\{\tilde{T}(0) > 4\}$ almost surely, then the causal target parameter is

identified as:

$$\frac{\mathbb{P}(T > 50|A = 1, \tilde{T} > 4) - \mathbb{P}(T > 50|A = 0, \tilde{T} > 4)}{\mathbb{P}(D^{\beta} = 1|A = 1, \tilde{T} > 4)}. \tag{19}$$

A total of 103 patients experienced an event or was censored before month 4. These patients are excluded from the population. An additional 395 patients missed visit 5 and we classify these patients as responders. Of the remaining 9125 patients in the population, 1446 experienced an event within the follow-up period.

We base the efficient plug-in estimator of the parameter on a survival random forest regression for the event endpoints and censoring times and an ensemble regression for the respondence indicator. Specifically, we rely on the survival random forest implementation of Wright and Ziegler (2017) and the super learner ensemble method of Van der Laan et al. (2007). The ensemble includes logistic regression, generalized additive model regression with integrated smoothness estimation and random forest regression. Each one-step estimate is cross-fitted using 5 splits to relax the restrictions on the model complexity. The results are displayed in Table 3. The estimated average treatment effect is 0.0209 with corresponding 95% confidence interval (0.004,0.038). If we are willing to assume that the stochastic exclusion restriction holds for a β-value of 0.4, the average treatment effect among responders is 0.0226, (0.004,0.41), which corresponds to a 8% increase. These results may reveal the performance of liraglutide relevant for the design of future treatment switching regimens.

Table 3 about here

6 | CONCLUDING REMARKS

In the setting of a randomized study with non-compliance in the active arm and with no access to the active treatment in the placebo arm, the estimand considered in this paper corresponds to the causal effect of treatment among patients that would comply to the active treatment if given the active treatment. In this setting the observed D is identical zero if $A = 0$. However, the results derived in this paper still hold in this case as the tangent space is unchanged, which is shown in the Appendix.

The stochastic exclusion restriction may seem like a strict assumption. However, as highlighted in our application, we can choose to be more or less conservative when classifying patients as non-responders. Moreover, even though the restriction is an untestable assumption it will induce bounds

that can be verified from the observed data, see Balke and Pearl (1997). Future work may focus on sensitivity analyses along the line of Díaz et al. (2018) investigating bounds for the causal bias. Since the exclusion restriction is untestable, the estimator we are considering should never be used as a primary outcome for any medical study. However, the estimator can serve as an inspiration for the design of (optimal) dynamic treatment regimen trials Chakraborty and Moodie (2013). These trials, without relying on any cross-world assumptions, can verify the benefit of the treatment for a subgroup of the patients.

Another important topic for future work is to study variable importance for a subset of the baseline variables. Results for targeting the best parametric least square approximation of the local average treatment effect have been established, see Ogburn et al. (2015), but an expression for the efficient influence along with properties of the associated one-step estimator has not been published. We also note that other identifying assumptions has been considered in the literature. Stuart and Jo (2015) considered the principal ignorability assumption: $T(0)$ and $D(1)$ are conditionally independent given W, which is sufficient to identify our considered target parameter. Since $T(0)$ and $D(1)$ cannot be jointly observed this is a so-called cross-world assumptions that is not verifiable based on the observed data. Another cross- world assumption that also makes identification possible is to require $T(0)$ and W being conditionally independent given $D(1)$ which is sometimes known as "auxiliary independence", see Jiang and Ding (2021). This is again an assumption that cannot be verified based on the observed data, but a situation compatible with the assumption, is when the included covariates effects on the outcome is via the biomarker response only. This is, however, unappealing for the situation we consider where we wish to include covariates that are strong predictors for the outcome as this will lead to increased precision of the proposed estimator. For more discussion on these assumptions, see Dukes et al. (2021).

The timing for assessing the post-baseline biomarker should occur relatively shortly after the initiation of treatment to avoid that the event of interest happens before the biomarker is measured. Even if a few events occur before the planned time point it is unlikely to influence the overall analysis much. If this can not be assumed one may define a further refined principal stratum such a restricting to those individuals that would survive this point in time if of on active treatment, see Bornkamp and Bermann (2019).

7 | ACKNOWLEDGEMENT

We acknowledge Novo Nordisk for providing the data for the LEADER trial Marso et al. (2016). Our analysis is conducted independently of the original study, and Novo Nordisk is not liable for any

errors or misinterpretations.

references

Angrist, J. D., Imbens, G. W. and Rubin, D. B. (1996) Identification of causal effects using instrumental variables. *Journal of the American statistical Association*, **91**, 444–455.

Balke, A. and Pearl, J. (1997) Bounds on treatment effects from studies with imperfect compliance. *Journal of the American Statistical Association*, **92**, 1171–1176.

Bickel, P. J., Klaassen, C. A., Bickel, P. J., Ritov, Y., Klaassen, J., Wellner, J. A. and Ritov, Y. (1993) *Efficient and adaptive estimation for semiparametric models*, vol. 4. Johns Hopkins University Press Baltimore.

Bornkamp, B. and Bermann, G. (2019) Estimating the treatment effect in a subgroup defined by an early post-baseline biomarker measurement in randomized clinical trials with time-to-event endpoint. *Statistics in Biopharmaceutical Research*.

Bornkamp, B., Rufibach, K., Lin, J., Liu, Y., Mehrotra, D. V., Roychoudhury, S., Schmidli, H., Shentu, Y. and Wolbers, M. (2021) Principal stratum strategy: potential role in drug development. *Pharmaceutical Statistics*, **20**, 737–751.

Chakraborty, B. and Moodie, E. E. (2013) Statistical methods for dynamic treatment regimes. *Springer-Verlag. doi*, **10**, 4–1.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W. and Robins, J. (2018) Double/debiased machine learning for treatment and structural parameters.

Díaz, I., Luedtke, A. R. and van der Laan, M. J. (2018) Sensitivity analysis. In *Targeted Learning in Data Science*, 511–522. Springer.

Didelez, V. and Sheehan, N. (2007) Mendelian randomization as an instrumental variable approach to causal inference. *Statistical methods in medical research*, **16**, 309–330.

Dukes, O., Van Lancker, K., Bornkamp, B., Heinzmann, D., Rufibach, K. and Wolbers, M. (2021) On identification of the principal stratum effect in patients who would comply if treated. *Statistics in Biopharmaceutical Research*, **13**, 508–510.

Egleston, B. L., Uzzo, R. G. and Wong, Y.-N. (2017) Latent class survival models linked by principal stratification to investigate heterogenous survival subgroups among individuals with early-stage kidney cancer. *Journal of the American Statistical Association*, **112**, 534–546.

Frangakis, C. E. and Rubin, D. B. (2002) Principal stratification in causal inference. *Biometrics*, **58**, 21–29.

Gilbert, P. B., Bosch, R. J. and Hudgens, M. G. (2003) Sensitivity analysis for the assessment of causal vaccine effects on viral load in hiv vaccine trials. *Biometrics*, **59**, 531–541.

Hines, O., Dukes, O., Diaz-Ordaz, K. and Vansteelandt, S. (2022) Demystifying statistical learning based on efficient influence functions. *The American Statistician*, 1–13.

Hirano, K., Imbens, G. W., Rubin, D. B. and Zhou, X.-H. (2000) Assessing the effect of an influenza vaccine in an encouragement design. *Biostatistics*, **1**, 69–88.

Imbens, G. W. and Rubin, D. B. (1997) Estimating outcome distributions for compliers in instrumental variables models. *The Review of Economic Studies*, **64**, 555–574.

Jiang, Z. and Ding, P. (2021) Identification of causal effects within principal strata using auxiliary variables. *Statistical Science*, **36**, 493–508.

Van der Laan, M. J., Laan, M. and Robins, J. M. (2003) *Unified methods for censored longitudinal data and causality*. Springer Science & Business Media.

Van der Laan, M. J., Polley, E. C. and Hubbard, A. E. (2007) Super learner. *Statistical applications in genetics and molecular biology*, **6**.

Larsen, K. G. and Josiassen, M. K. (2020) A new principal stratum estimand investigating the treatment effect in patients who would comply, if treated with a specific treatment. *Statistics in Biopharmaceutical Research*, **12**, 29–38.

Lee, Y., Kennedy, E. H. and Mitra, N. (2023) Doubly robust nonparametric instrumental variable estimators for survival outcomes. *Biostatistics*, **24**, 518–537.

Loeys, T. and Goetghebeur, E. (2003) A causal proportional hazards estimator for the effect of treatment actually received in a randomized trial with all-or-nothing compliance. *Biometrics*, **59**, 100–105.

Magnusson, B. P., Schmidli, H., Rouyrre, N. and Scharfstein, D. O. (2019) Bayesian inference for a principal stratum estimand to assess the treatment effect in a subgroup characterized by postrandomization event occurrence. *Statistics in Medicine*, **38**, 4761–4771.

Marso, S. P., Daniels, G. H., Brown-Frandsen, K., Kristensen, P., Mann, J. F., Nauck, M. A., Nissen, S. E., Pocock, S., Poulter, N. R., Ravn, L. S. et al. (2016) Liraglutide and cardiovascular outcomes in type 2 diabetes. *New England Journal of Medicine*, **375**, 311–322.

Ogburn, E. L., Rotnitzky, A. and Robins, J. M. (2015) Doubly robust estimation of the local average treatment effect curve. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **77**, 373–396.

Ridker, P. M., Everett, B. M., Thuren, T., MacFadyen, J. G., Chang, W. H., Ballantyne, C., Fonseca, F., Nicolau, J., Koenig, W., Anker, S. D. et al. (2017) Antiinflammatory therapy with canakinumab for atherosclerotic disease. *New England journal of medicine*, **377**, 1119–1131.

Rytgaard, H. C. W., Eriksson, F. and van der Laan, M. (2021) Estimation of time-specific intervention effects on continuously distributed time-to-event outcomes by targeted maximum likelihood estimation. *arXiv preprint arXiv:2106.11009*.

Stuart, E. A. and Jo, B. (2015) Assessing the sensitivity of methods for estimating principal causal effects. *Statistical methods in medical research*, **24**, 657–674.

Tsiatis, A. A. (2006) Semiparametric theory and missing data.

Van der Vaart, A. W. (2000) *Asymptotic statistics*, vol. 3. Cambridge university press.

Westling, T., Luedtke, A., Gilbert, P. and Carone, M. (2021) Inference for treatment-specific survival curves using machine learning. *arXiv preprint arXiv:2106.06602*.

Wright, M. N. and Ziegler, A. (2017) ranger: A fast implementation of random forests for high dimensional data in c++ and r. *Journal of Statistical Software*, **77**, 1–17.

Corresponding author: Andreas Nordland, Section of Biostatistics, University of Copenhagen, Øster Farimagsgade 5, DK-1014 Copenhagen, Denmark. email: anno@sund.ku.dk.

Appendix

Generally, any influence function for a parameter $\Psi(P_0)$ can be written as $\psi(P_0)(O) = \phi(P_0)(O) - \Psi(P_0)$ for some function $\phi(P_0)(O)$. We will use the notation $P_n V = n^{-1} \sum_{i=1}^n V_i$ for n independent and identically distributed variables $(V_i)_{i\in\{1,\dots,n\}}$ and $PV = \int V dP$. Define the one-step estimator $\Psi_n(\hat{P}_n) = P_n\phi(\hat{P}_n)(O)$ based on a fitted model $\hat{P}_n$. Let P^* be the limiting model of $\hat{P}_n$ possibly different from P_0. Define the *second order remainder* as

$$R(\hat{P}_n, P_0) = P_0\phi(\hat{P}_n)(O) - \Psi(P_0). \tag{20}$$

Then it is simple to show that

$$\begin{aligned}\Psi_n(\hat{P}_n) - \Psi(P_0) &= \{P_n - P_0\}\,\phi(P^*)(O) \\ &\quad + \{P_n - P_0\}\left\{\phi(\hat{P}_n)(O) - \phi(P^*)(O)\right\} &(21)\\ &\quad + R(\hat{P}_n, P_0), &(22)\end{aligned}$$

where we denote (21) as the *empirical process remainder*. If both the empirical process remainder and the second order remainder are $o_{P_0}(1)$, the one-step estimator will be consistent. If in addition to that, both the empirical process remainder and the second order remainder are $o_{P_0}(n^{-1/2})$, the one-step estimator $\Psi_n(\hat{P}_n)$ has influence function $\phi(P^*)(O) - \mathbb{E}\left[\phi(P^*)(O)\right]$. Consequently, for

$\Psi(P_0) \in \mathbb{R}$,

$$n^{1/2}(\Psi_n(\hat{P}_n) - \Psi(P_0)) \xrightarrow{\mathcal{D}} \mathcal{N}\left(0, \mathbb{E}\left[\phi(P^*)(O) - \mathbb{E}\{\phi(P^*)(O)\}\right]^2\right), \tag{23}$$

which allow us to make inference on the estimator. In practice, the asymptotic variance of the one-step estimator can be estimated by $\hat{\sigma}_n^2 = n^{-1} P_n \left\{\phi(\hat{P}_n)(O) - P_n\phi(\hat{P}_n)(O)\right\}^2$. Furthermore, if $\psi(P_0)(O)$ is the efficient influence function and $P^* = P_0$, the one-step estimator will be asymptotically efficient, i.e., the one-step estimator asymptotically achieves the Cramer-Rao lower bound, see Van der Vaart (2000) Section 25.3. To prove that empirical process remainder, (21), is $o_{P_0}(n^{-1/2})$, it is enough to show that $\phi(\hat{P}_n)$ falls in a Donsker class with probability tending to one and that $P_0(\phi(\hat{P}_n)(O) - \phi(P^*)(O))^2 = o_{P_0}(1)$.

Suppose that we have a vector-valued estimator $\Psi_n \in \mathbb{R}^m$ of a parameter $\Psi(P_0) \in \mathbb{R}^m$ with influence function $\psi(P_0) : O \to \mathbb{R}^m$. Then, for a differentiable function $f : \mathbb{R}^m \to \mathbb{R}$, the estimator $f(\Psi_n)$ has influence function

$$\nabla f(\Psi(P_0))^T \cdot \psi(P_0),$$

see Van der Vaart (2000) Section 25.7. Thus, given influence functions $\psi^D(P_0)$, $\psi^1(P_0)$ and $\psi^0(P_0)$ for estimators $\Psi_n^D(\hat{P}_n)$, $\Psi_n^1(\hat{P}_n)$ and $\Psi_n^0(\hat{P}_n)$, the plug-in estimator

$$\Psi_n(\hat{P}_n) = \frac{\Psi_n^1(\hat{P}_n) - \Psi_n^0(\hat{P}_n)}{\Psi_n^D(\hat{P}_n)} \tag{24}$$

has influence function

$$\psi(P_0)(O) = \frac{1}{\Psi^D(P_0)}\left(\psi^1(P_0)(O) - \psi^0(P_0)(O) - \frac{\Psi^1(P_0) - \Psi^0(P_0)}{\Psi^D(P_0)}\psi^D(P_0)(O)\right). \tag{25}$$

If the influence functions $\psi^D(P_0)$, $\psi^1(P_0)$ and $\psi^0(P_0)$ are efficient, then so is $\psi(P_0)(O)$.

| Tangent space

We start by calculating the tangent space in the setting where D is always observed corresponding the main application in our paper where is the post-treatment biomarker response. Let $Z :=$

(W, A, D, Y). Consider the Hilbert space $\mathcal{H}$ of $L^2(\mathbb{P})$ zero mean functions of Z endowed with covariance inner product. Due to randomization the tangent space, which is a subspace of $\mathcal{H}$, is given by following orthogonal decomposition

$$\mathcal{T} = \mathcal{T}_W \oplus \mathcal{T}_D \oplus \mathcal{T}_Y$$

where

$$\begin{aligned}
\mathcal{T}_W &= \{h(W) \in \mathcal{H} : \mathbb{E}[h(W)] = 0\} \\
\mathcal{T}_D &= \{h(D, A, W) \in \mathcal{H} : \mathbb{E}[h(D, A, W) \mid A, W] = 0\} \\
\mathcal{T}_Y &= \{h(Y, D, A, W) \in \mathcal{H} : \mathbb{E}[h(Y, D, A, W) \mid D, A, W] = 0\}.
\end{aligned}$$

Thus, it is clear that the orthogonal complement to the tangent set is given by

$$\begin{aligned}
\mathcal{T}^{\perp} &= \{h(A, W) \in \mathcal{H} : \mathbb{E}[h(A, W) \mid W] = 0\} \\
&= \left\{(A - \delta)h_*(W) : \mathbb{E}\left[h_*(W)^2\right] < \infty\right\}. \qquad (26)
\end{aligned}$$

Note that (26) holds since A is binary and we know that $\mathbb{E}[h(A, W) \mid W] = 0$. We now derive the tangent space in the setting where D is only observed if $A = 1$ mirroring the situation with a randomized study where there is non-compliance and where patients randomized to placebo do not have access to the active treatment. We denote the (full data) tangent space by $\mathcal{T}$ and will now argue that its orthogonal complement is again given by (26). In this case, data can be written as $Z = \{Y, (A, AD), W)\}$ with $Y = I(T \leq t)$ in the survival setting. Let

$$\begin{aligned}
\mathcal{T}_1 &= \{\alpha(Z) : \ E\{\alpha(Z)|(A, AD), W\} = 0\} \\
\mathcal{T}_2^* &= \{\alpha\{(A, AD), W\} : \ E\{\alpha\{(A, AD), W\}|W\} = 0\} \\
\mathcal{T}_3 &= \{\alpha(W) : \ E\{\alpha(W)\} = 0\},
\end{aligned}$$

The three tangent spaces in the latter display are mutually orthogonal and

$$\mathcal{H} = \mathcal{T}_1 \oplus \mathcal{T}_2^* \oplus \mathcal{T}_3$$

where $\mathcal{H}$ is the Hilbert-space of zero-mean functions $\alpha(Z)$. Let

$$\mathcal{T}_2 = \left\{ Ah(W) \left[I(A = 1, D = 0) - P(D = 0 | A = 1, W) \right] \right\}$$

Then it is easy to see that the tangent space is given by

$$\mathcal{T} = \mathcal{T}_1 \oplus \mathcal{T}_2 \oplus \mathcal{T}_3$$

and from the decomposition of the full Hilbert space we see that

$$(\mathcal{T}_1 \oplus \mathcal{T}_3)^{\perp} = \mathcal{T}_2^{*}.$$

Further, since

$$\mathcal{T}^{\perp} = (\mathcal{T}_1 \oplus \mathcal{T}_3)^{\perp} \cap \mathcal{T}_2^{\perp},$$

we see that $\mathcal{T}^{\perp}$ consist of elements in $\mathcal{T}_2^{*}$ that are also perpendicular to $\mathcal{T}_2$. After a little algebra it is seen that this is again given by (26).

| Derivation of efficient influence function and remainder term

Let $Z = (W, A, D, T)$ denote the full data. In the full data case, an influence function for $\Psi^a(P_0) = \mathbb{E}[I\{T \leq \tau\} | A = a]$ is given by

$$\psi^a(Z)(P_0) = \frac{I\{A = a\}}{g_0(a)} \left(I\{T \leq \tau\} - \Psi^a(P_0) \right)$$

The efficient influence function is now given by the projection of the gradient onto the tangent space. The orthogonal complement to the tangent space is given by the space $\{(A - \delta)h(W) : \mathbb{E}[h(W)^2] <$

$\infty\}$. The resulting projection yields that the full data efficient influence function is given by

$$\tilde{\psi}^a(Z)(P_0) = \frac{I\{A=a\}}{g_0(a)}\{I\{T \leq \tau\} - F_0(\tau|A=a,W)\} + F_0(\tau|A=a,W) - \Psi^a(P_0),$$

where

$$F_0(\tau|A=a,W) = \sum_{d=0,1}\left\{\int_0^\tau S_0(u|W,a,d)d\Lambda_0(u|W,a,d)\right\}h_0(d|W,a) = \mathbb{P}(T \leq \tau|A=a,W).$$

Let $X = (W,A,D)$. In the observed data case formula (10.76) in Tsiatis (2006) yields that the observed data efficient influence function is given by

$$\begin{aligned}&\tilde{\psi}^a(O)(P_0)\\ =&\frac{\Delta(\tau)}{S_0^c(\min(\tilde{T},\tau)|X)}\tilde{\psi}^a(Z)(P_0) + \int_0^\tau \frac{\mathbb{E}\left[\tilde{\psi}^a(Z)(P_0)|T \geq u, X\right]}{S_0^c(u|X)}dM_0^c(u|X),\end{aligned}$$

where $\Delta(\tau) = I\{C > \min(T,\tau)\}$ and

$$M_0^c(s|X) = I\{\tilde{T} \leq s, \Delta = 0\} - \int_0^s I\{\tilde{T} \geq u\}d\Lambda_0^c(u|X).$$

A similar result can also be found in Example 1.12 in (Van der Laan et al., 2003). Because

$$\mathbb{E}\left[I\{T \leq \tau\}|T \geq u, X\right] = I\{u \leq \tau\}\frac{S_0(u|X) - S_0(\tau|X)}{S_0(u|X)},$$

the observed data efficient influence function is given by

$$\begin{aligned}\tilde{\psi}^a(O)(P_0) =& \frac{I\{A=a\}}{g_0(a)}\left(\frac{\Delta(\tau)}{S_0^c(\min(\tilde{T},t)|X)}I\{\tilde{T} \leq \tau\} + \int_0^\tau \frac{S_0(u|X) - S_0(\tau|X)}{S_0(u|X)S_0^c(u|X)}dM_0^c(u|X)\right)\\ &+ \left(1 - \frac{I\{A=a\}}{g_0(a)}\right)F_0(\tau|A=a,W) - \Psi^a(P_0).\end{aligned} \tag{27}$$

For a fixed model P of P_0 we are interested in finding an expression for the second order remainder. For that purpose we notice that the first term of the parentheses of (27) can be rewritten as

$$\begin{aligned}\frac{\Delta(\tau)}{S_0^c(\min(\tilde{T},t)|X)}I\{\tilde{T}\leq\tau\} =& I\{T\leq\tau\}+\left\{\frac{\Delta(\tau)}{S_0^c(\min(T,\tau)|X)}-1\right\}I\{T\leq\tau\}\\ =& I\{T\leq\tau\}-\int_0^\tau \frac{I\{T\leq\tau\}}{S_0^c(u|X)}dM_0^c(u|X).\end{aligned}$$

The first equality holds because if $C > \min(T,\tau)$ and $\min(T,C)\leq\tau$ then $C > T$. The second equality holds because

$$\begin{aligned}\int_0^\tau \frac{I\{T\leq\tau\}}{S_0^c(u|X)}dM_0^c(u|X) &= I\{T\leq\tau\}\left\{\frac{(1-\Delta)I\{\tilde{T}\leq\tau\}}{S_0^c(\tilde{T}|X)}-\int_0^{\min(\tilde{T},\tau)}\frac{1}{S_0^c(u|X)}d\Lambda_0^c(u|X)\right\}\\ &= I\{T\leq\tau\}\left\{\frac{(1-\Delta)I\{\tilde{T}\leq\tau\}}{S_0^c(\tilde{T}|X)}-\frac{1}{S_0^c(\min(\tilde{T},\tau))}+1\right\}\\ &= I\{T\leq\tau\}\left\{\frac{1-\Delta(\tau)}{S_0^c(\min(\tilde{T},\tau)|X)}-\frac{1}{S_0^c(\min(\tilde{T},\tau))}+1\right\}\\ &= I\{T\leq\tau\}\left\{\frac{-\Delta(\tau)}{S_0^c(\min(\tilde{T},\tau)|X)}+1\right\}.\end{aligned}$$

Thus, the efficient influence function is also equal to

$$\begin{aligned}\tilde{\psi}^a(O)(P_0) =& \frac{I\{A=a\}}{g_0(a)}\left(I\{T\leq\tau\}-\int_0^\tau\left\{\frac{I\{T\leq\tau\}}{S_0^c(u|X)}-\frac{S_0(u|X)-S_0(\tau|X)}{S_0(u|X)S_0^c(u|X)}\right\}dM_0^c(u|X)\right)\\ &+\left(1-\frac{I\{A=a\}}{g_0(a)}\right)F_0(\tau|A=a,W)-\Psi^a(P_0).\end{aligned}$$

By definition, the second order remainder is now given by

$$\begin{aligned}&R^a(P,P_0)\\ =&-\mathbb{E}\left[\frac{I\{A=a\}}{g_0(a)}\int_0^\tau\left\{\frac{I\{T\leq\tau\}}{S^c(u|X)}-\frac{S(u|X)-S(\tau|X)}{S(u|X)S^c(u|X)}\right\}dM^c(u|X)\right]\\ &+\mathbb{E}\left[\left(1-\frac{I\{A=a\}}{g_0(a)}\right)F(\tau|A=a,W)\right]\\ &+\mathbb{E}\left[\frac{I\{A=a\}}{g_0(a)}I\{T\leq\tau\}-\Psi^a(P_0)\right]\\ =&-\mathbb{E}\left[\frac{I\{A=a\}}{g_0(a)}\int_0^\tau\left\{\frac{I\{T\leq\tau\}}{S^c(u|X)}-\frac{S(u|X)-S(\tau|X)}{S(u|X)S^c(u|X)}\right\}dM^c(u|X)\right].\end{aligned}$$

We aim to use the tower property in order to replace $I\{T \leq \tau\}$ in the above expression. For this purpose we note that

$$\begin{aligned}
&\mathbb{E}\left[\int_0^\tau \frac{I\{T \leq \tau\}}{S^c(u|X)} dN^c(u)\right] \\
=&\mathbb{E}\left[I\{T \leq \tau\} \frac{(1-\Delta) I\{\tilde{T} \leq \tau\}}{S^c(\tilde{T}|X)}\right] \\
=&\mathbb{E}\left[\mathbb{E}\left(I\{T \leq \tau\}|X, \tilde{T}, \Delta = 0\right) \frac{1-\Delta}{S^c(\tilde{T}|X)}\right] \\
=&\mathbb{E}\left[\mathbb{E}\left(I\{T \leq \tau\}|X, T > C, C\right) \frac{1-\Delta}{S^c(\tilde{T}|X)}\right] \\
=&\mathbb{E}\left[I\{\tilde{T} \leq \tau\} \frac{S_0(\tilde{T}|X) - S_0(\tau|X)}{S_0(\tilde{T}|X)} \frac{1-\Delta}{S^c(\tilde{T}|X)}\right],
\end{aligned}$$

and

$$\begin{aligned}
&\mathbb{E}\left[\int_0^\tau \frac{I\{T \leq \tau\}}{S^c(u|X)} I\{\tilde{T} \geq u\} d\Lambda^c(u|X)\right] \\
=&\mathbb{E}\left[\int_0^\tau \frac{\mathbb{E}\left[I\{T \leq \tau\}|X, \tilde{T} \geq u\right]}{S^c(u|X)} I\{\tilde{T} \geq u\} d\Lambda^c(u|X)\right] \\
=&\mathbb{E}\left[\int_0^\tau \frac{\mathbb{E}\left[I\{T \leq \tau\}|X, T \geq u\right]}{S^c(u|X)} I\{\tilde{T} \geq u\} d\Lambda^c(u|X)\right] \\
=&\mathbb{E}\left[\int_0^\tau \frac{S_0(u|X) - S_0(\tau|X)}{S_0(u|X) S^c(u|X)} I\{\tilde{T} \geq u\} d\Lambda^c(u|X)\right].
\end{aligned}$$

Combining the two results yields that we can rewrite the second-order remainder as

$$\begin{aligned}
&-R^a(P, P_0) \\
=&\mathbb{E}\left[\frac{I\{A=a\}}{g_0(a)} \int_0^\tau \left\{\frac{S_0(u|X) - S_0(\tau|X)}{S_0(u|X)} - \frac{S(u|X) - S(\tau|X)}{S(u|X)}\right\} \frac{1}{S^c(u|X)} dM^c(u|X)\right] \\
=&\mathbb{E}\left[\frac{I\{A=a\}}{g_0(a)} \int_0^\tau \left\{\frac{S_0(u|X) - S_0(\tau|X)}{S_0(u|X)} - \frac{S(u|X) - S(\tau|X)}{S(u|X)}\right\} \frac{E[I\{\tilde{T} \geq u\}|X]}{S^c(u|X)} \left\{d\Lambda_0^c(u|X) - d\Lambda^c(u|X)\right\}\right] \\
=&\mathbb{E}\left[\frac{I\{A=a\}}{g_0(a)} \int_0^\tau \left\{\frac{S_0(u|X) - S_0(\tau|X)}{S_0(u|X)} - \frac{S(u|X) - S(\tau|X)}{S(u|X)}\right\} \frac{S_0(u|X) S_0^c(u|X)}{S^c(u|X)} \left\{d\Lambda_0^c(u|X) - d\Lambda^c(u|X)\right\}\right] \\
=&\mathbb{E}\left[\frac{I\{A=a\}}{g_0(a)} \int_0^\tau \left\{\frac{S(\tau|X)}{S(u|X)} - \frac{S_0(\tau|X)}{S_0(u|X)}\right\} \frac{S_0(u|X) S_0^c(u|X)}{S^c(u|X)} \left\{d\Lambda_0^c(u|X) - d\Lambda^c(u|X)\right\}\right].
\end{aligned}$$

Finally, note that

$$\begin{aligned}\frac{S(\tau|X)}{S(u|X)} - \frac{S_0(\tau|X)}{S_0(u|X)} &= \frac{S(\tau|X)S_0(u|X) - S_0(\tau|X)S(u|X)}{S(u|X)S_0(u|X)} \\ &= \frac{S(\tau|X)S_0(u|X) - \left[S_0(\tau|X) + S(\tau|X) - S(\tau|X)\right]S(u|X)}{S(u|X)S_0(u|X)} \\ &= \frac{S(\tau|X)\left[S_0(u|X) - S(u|X)\right] - \left[S_0(\tau|X) - S(\tau|X)\right]S(u|X)}{S(u|X)S_0(u|X)} \\ &= \frac{S(\tau|X)\left[S_0(u|X) - S(u|X)\right]}{S(u|X)S_0(u|X)} - \frac{\left[S_0(\tau|X) - S(\tau|X)\right]}{S_0(u|X)}.\end{aligned}$$

Hence,

$$\begin{aligned}-R^a(P, P_0) = \mathbb{E}\Bigg[\frac{I\{A = a\}}{g_0(a)} \int_0^\tau \Bigg\{\frac{S(\tau|X)}{S(u|X)}\left[S_0(u|X) - S(u|X)\right] - \left[S_0(\tau|X) - S(\tau|X)\right]\Bigg\} \\ \times \frac{S_0^c(u|X)}{S^c(u|X)}\left\{d\Lambda_0^c(u|X) - d\Lambda^c(u|X)\right\}\Bigg].\end{aligned}$$

TABLE 1 Simulation study: Cross-fitted efficient one-step estimation of the average treatment effect among responders with a binary outcome. The study is based on 1000 replications with a sample size of 1000 using 5 fold cross-fitting. The efficient one-step estimate is described in Section 3.1. The empirical mean plug-in estimate is based on plug-in of the empirical mean estimates in Equation (3).

	Mean	SD	Bias	RMSE	SE	Coverage
Empirical mean plug-in estimate	-0.103	0.039	0.001	0.039	0.038	0.943
Efficient one-step estimate	-0.104	0.035	0.001	0.035	0.035	0.945

TABLE 2 Simulation study: Efficient one-step estimation of the average treatment effect among responders with a time-to-event outcome under right censoring. The estimation procedure is replicated 1000 times with a sample size of 1000 (500 randomized to each group). The cumulative hazard functions $\Lambda(\cdot|X)$ and $\Lambda^c(\cdot|X)$ are fitted using Cox proportional hazards models. The model inputs X varies accordingly for the different cases. The response indicator model H inputs W if and only if X includes W. The final estimator KM is the plug-in estimate based on the Kaplan-Meier estimate for the treatment effect and the empirical probability estimate for the response.

	Mean	SD	Bias	RMSE	SE	Coverage
$X = (W, A, D)$	-0.1708	0.0518	0.0001	0.0518	0.0514	0.9530
$X = (A, D)$	-0.1700	0.0575	0.0009	0.0575	0.0583	0.9510
$X = (W, A)$	-0.1948	0.0462	-0.0239	0.0520	0.0469	0.9190
KM	-0.1901	0.0539	-0.0192	0.0572	0.0528	0.9150

TABLE 3 Estimates of (25) for various values of β. A β-value of $-\infty$ corresponds to everyone being classified as a responder, meaning that the target parameter equals the usual average treatment effect.

beta	Estimate	Std.Err	2.5%	97.5%	P-value
$-\infty$	0.0209	0.0086	0.0042	0.0377	0.0145
0	0.0213	0.0089	0.0038	0.0389	0.0170
0.2	0.0224	0.0091	0.0045	0.0403	0.0140
0.4	0.0226	0.0093	0.0043	0.0408	0.0152
0.5	0.0242	0.0095	0.0055	0.0429	0.0112
0.6	0.0241	0.0097	0.0052	0.0430	0.0124