Improving the Efficiency of Mortality Estimators in Resampling Designs with Undersmoothed Highly Adaptive Lasso

by

Kirsten Landisedel

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Arts

in

Biostatistics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Mark van der Laan, Chair
Professor Maya Petersen
Professor Laura Balzer

Summer 2023

Improving the Efficiency of Mortality Estimators in Resampling Designs with Undersmoothed Highly Adaptive Lasso

Abstract

Improving the Efficiency of Mortality Estimators in Resampling Designs with Undersmoothed Highly Adaptive Lasso

by

Kirsten Landisedel

Master of Arts in Biostatistics

University of California, Berkeley

Professor Mark van der Laan, Chair

Resampling designs have proven invaluable in resource-limited settings where substantial loss to follow-up (LTFU) impedes accurate estimation of mortality among patients with HIV. Mortality rates are often higher among patients who cease reporting to clinic visits, leaving a large potential for bias on the table. One paper reported a jump in estimated mortality from 2.7% (calculated from passively recorded deaths only) to 7.1% (calculated from passively recorded deaths combined with deaths discovered through resampling) [1]. While techniques exist to incorporate resampled data into mortality estimation procedures, insufficient attention has been paid to constructing truly efficient and unbiased estimators for mortality. We present and compare three different estimators in the context of resampling: (1) an inverse probability of censoring weighted (IPCW) estimator, (2) an IPCW-HAL estimator which uses undersmoothed Highly Adaptive Lasso (HAL) estimators to estimate the relevant propensity scores, and (3) a plug-in estimator which estimates each non-intervention factor of the likelihood with undersmoothed HAL (though we leave simulations for estimator 3 to future work). Ultimately, we advocate for the incorporation of undersmoothed HAL when constructing IPW estimators as our simulation studies demonstrate an approximate 30% reduction in variance of IPCW estimators when using undersmoothed HAL to estimate the propensity sores relative to using known probabilities.

Contents

Contents **ii**

List of Figures **iii**

List of Tables **iv**

1 Introduction **1**
1.1 Literature Review: Resampling Designs 2
1.2 Background: Highly Adaptive Lasso 4
1.3 Background: Undersmoothing HAL 6
1.4 Background: RAL Estimators & Efficiency Theory 7

2 Methodology **9**
2.1 Defining the Estimation Problem 9
2.2 Estimator 1: IPCW 15
2.3 Estimator 2: IPCW-HAL 15
2.4 Estimator 3: Plug-in Estimator Based on Undersmoothed HAL 17

3 Simulations & Results **20**
3.1 Simulation Results for IPCW vs IPCW + HAL Estimators 21

4 Discussion **30**

Bibliography **32**

5 Appendix **34**

List of Figures

3.1 Interval plots for the sampling distribution of point estimates from simulations on data of sample sizes 1k and 10k to assess performance of the IPCW estimator using known probabilities for weights versus IPCW+HAL estimators with different degrees of undersmoothing (Independent Resampling). 24
3.2 Bias of Estimators in Simulation 2 (Independent Resampling). From left to right: IPCW without HAL, IPCW+HAL with no undersmoothing, IPCW+HAL with increasing amounts of undersmoothing. 25
3.3 Interval plots for the sampling distribution of point estimates from simulations on data of sample sizes 1k and 10k to assess performance of IPCW estimator using known probabilities for weights versus IPCW+HAL estimators with different degrees of undersmoothing (Dependent Resampling). 28
3.4 Bias of Estimators in Simulation 2 (Dependent Resampling). From left to right: IPCW without HAL, IPCW+HAL with no undersmoothing, IPCW+HAL with increasing amounts of undersmoothing. 29

List of Tables

3.1 Simulation results for IPCW estimator which uses known probabilities for weights (Independent Resampling). These results were based on 10k simulations of independently drawn data. 21

3.2 Simulation results for IPCW-HAL estimators across different levels of undersmoothing applies to data sets of size 1k (Independent Resampling). These results were based on 1k simulations of independently drawn data. The λ column corresponds to the parameter used in the HAL fit, which controls the amount of undersmoothing (where λ_{CV} corresponds to no undersmoothing and the smaller values of λ correspond to increased amounts of undersmoothing). 22

3.3 Simulation results for IPCW-HAL estimators across different levels of undersmoothing applies to data sets of size 10k (Independent Resampling). These results were based on 1k simulations of independently drawn data. The λ column corresponds to the parameter used in the HAL fit, which controls the amount of undersmoothing (where λ_{CV} corresponds to no undersmoothing and the smaller values of λ correspond to increased amounts of undersmoothing). 23

3.4 Simulation results for the IPCW estimator with known probabilities for weights (Dependent Resampling). These results were based on 10k simulations of independently drawn data. 26

3.5 Simulation results for IPCW-HAL estimators across different levels of undersmoothing applies to data sets of size 1k (Dependent Resampling). These results were based on 1k simulations of independently drawn data. The λ column corresponds to the parameter used in the HAL fit, which controls the amount of undersmoothing (where λ_{CV} corresponds to no undersmoothing and the smaller values of λ correspond to increased amounts of undersmoothing). 26

3.6 Simulation results for IPCW-HAL estimators across different levels of undersmoothing applies to data sets of size 10k (Dependent Resampling). These results were based on 1k simulations of independently drawn data. The λ column corresponds to the parameter used in the HAL fit, which controls the amount of undersmoothing (where λ_{CV} corresponds to no undersmoothing and the smaller values of λ correspond to increased amounts of undersmoothing). 27

Chapter 1

Introduction

According to the WHO, there were an estimated 38.4 million [33.9–43.8 million] people living with HIV at the end of 2021, two thirds of whom (25.6 million) are in the WHO African Region. As global public health experts set their sights on the elimination of HIV, having a reliable way to assess the efficacy of current standards of care is vital. Currently, mortality is the most important indicator of success for HIV treatment programs, yet routine monitoring fails to capture most deaths in resource-limited settings [2]. Studies have shown that estimating mortality from passively recorded deaths alone may underestimate mortality by 80% [3]. High loss to follow-up rates are a major impediment to researchers' ability to accurately quantify mortality as many patients drop out of these studies before their survival outcome is known. Mortality rates are often higher among loss to follow-up patients in this context, leading to underestimation of the true mortality rate without proper adjustment. Resampling, or extensive tracking to ascertain the vital status of a subset of patients with unknown outcome, has proven an invaluable tool to decrease bias and improve precision. One resampling-based study in a similar context reported a jump in estimated mortality of 2.7% to 7.1% after engaging in resampling [2].

Once resampled data has been collected, this new information must be incorporated into the mortality estimation procedure. While Kaplan-Meier and IPCW are commonly used for this purpose, these approaches may not be optimal without further augmentation [3] [4].

It has recently been shown that inverse probability weighted estimators (IPW) which use undersmoothed Highly Adaptive Lasso (HAL) estimators for their propensity scores are asymptotically linear and can have variance converging to the nonparametric efficiency bound [5]. This approach, however, has not been implemented or evaluated for this application. These estimators do not require derivation of the efficient influence function for implementation purposes (though insight about efficiency can be gained from doing so) or specification of a model for the outcome regression. Undersmoothed HAL estimators can also be used to produce asymptotically efficient plug-in estimators for the target parameter of interest while preserving HAL's faster than $n^{-\frac{1}{4}}$ rate of convergence [6]. In this paper, we will present and compare the efficiency of three estimators for mortality in resampling designs (1) an IPCW estimator, (2) an IPCW-HAL estimator which uses undersmoothed highly adaptive lasso estimators for the propensity scores, and (3) a plug-in estimator which relies on estimating all non-intervention factors of the likelihood using undersmoothed highly adaptive lasso estimators (simulations for estimator 3 are left to future work). Our goal is to demonstrate that incorporating HAL estimators into standard estimation procedures can improve efficiency and outperform existing estimators in resampling designs.

For the remainder of this paper, we use "right censoring" to mean administrative right censoring by end of study (end of study time varies across individuals). We will use "loss to follow-up" (LTFU) to refer to people who cease coming into the clinic before their study endpoint (thus, we do not know the outcome status for these individuals). There is no observed time as which a participant becomes LTFU for this event is unobserved; missed visits to not necessarily indicate LTFU as patients may return to the clinic at a later date. Survival time is independent of right censoring time but not independent of LTFU due to shared common causes.

The rest of the paper proceeds as follows. Section 1.1 contains a literature review that evaluates the strengths and weakness of existing approaches to estimation in resampling designs. Section 1.2 provides background on the Highly Adaptive Lasso, Section 1.3 adds additional information about undersmoothing, and Section 1.4 provides information about constructing efficient estimators. Chapter 2 then follows the "roadmap" for statistical learning in order to define the estimation problem [7]. Subsequently, the three proposed estimators are presented; their expected behavior and efficiency are discussed. Chapter 3 presents results for our three estimators on simulated data which mimics data from HIV treatment efficacy studies as in [2]. Lastly, Chapter 4 discusses the results of our analysis as well as areas for improvement.

1.1 Literature Review: Resampling Designs

The non-parametric maximum likelihood estimator (NPMLE) of the survival function is known to be equivalent to the Kaplan-Meier estimator when survival time is noninformatively right censored, making it one of the most commonly used estimators for failure time in survival analysis. Kaplan-Meier estimators remain a popular choice for analyzing data from resampling designs despite several complications that arise in this context [3] [2] [8].

First, it is not immediately clear how to combine data from observed deaths in stage one with data from resampled deaths in stage two. One paper compares different implementations of Kaplan-Meier in this context: (1) a KM estimator based on observed deaths only, (2) a KM estimator based on all deaths (those passively recorded and those discovered through resampling), and (3) a weighted average of Kaplan-Meier estimators in dropout and non-dropout groups [3]. Since risk factors for patient dropout coincide with risk factors for mortality, KM estimator (1) will be biased. Simple pooling of the deaths among dropout and non-dropout groups as in KM estimator (2) will also lead to bias since mortality rates are much higher among resampled (once dropout) patients, and resampling does not cover all dropout patients. KM estimator (3) gets us closest to the truth by adjusting for the relative size of dropout and non-dropout groups; however, this method has also been show to be biased by other sources [9]. In response to this bias, Frangakis and Rubin proposed using a weighted average of the hazard of death between dropout and non-dropout groups to estimate survival [9]. The seminal method presented in this paper remains a popular choice, with somewhat more recent papers continuing to advocate for its use over Kaplan-Meier [3]. While this approach may address known bias, it is inefficient and puts unrealistic restrictions on the problem [10]. The first of these restrictions requires resampling to be completely random. Random resampling is likely a suboptimal strategy, and prior literature suggests that more efficient designs rely on participants' covariate history to set resampling probabilities [11]. The second restriction requires that everyone chosen for resampling is successfully located, which is highly problematic as previous studies have reported an inability to acquire vital status information on 25-30% of patients chosen for resampling [1] [3]. Our work thus aims to provide an avenue for constructing efficient estimators for mortality within a flexible framework which can handle these complications without needing to impose further untenable assumptions.

Kaplan-Meier estimators also require a censoring time or an observed outcome time for each person. In this context we have right censoring times for patients at their end of study point τ. However, we do not have LTFU censoring times as some patients drop out of the study without warning befor their end of study; for these dropout patients, we have neither a LTFU censoring time nor an observed outcome time. It is not known whether these patients will return to the clinic at a future time point. Attempts have been made to construct artificial LTFU time for dropout patients based on whether or not they return to the clinic, but this practice can cause bias by violating the "no informative right censoring" assumption needed for KM [3].

Due to suspected dependence between censoring and survival, Robins and Finkelstein present inverse probability of censoring weighted (IPCW) versions of Kaplan-Meier and Cox partial likelihood estimators [4]. They recommend using Cox proportional hazard models to estimate the censoring mechanisms. Inclusion of IPCW weights can relax the assumption of strict independence between censoring and survival. Using Cox proportional hazard models for the censoring mechanism imposes assumptions about the effect of predictors on the hazard function; thus, we believe we can improve upon this method by using nonparametric estimators for the hazard of right censoring (by end of study) when implementing our IPCW and IPCW-HAL estimators.

Li and Tseng propose an estimator based on the Nelson-Aalen estimator of the cumulative hazard [8]. Their method is non-parametric and estimates the second stage survival function while borrowing information from stage one. They claim that their estimator corrects for bias in the first stage while improving upon the efficiency of the Kaplan-Meier estimator based on second stage data alone.

More recent work has proposed an inverse probability of censoring weighted targeted maximum likelihood estimator (IPCW-TMLE) for two-stage designs [12]. TMLE is a general procedure for estimating target parameters of data-generating distributions which involves two steps: obtaining initial estimates of the relevant portions of the data-generating distribution and then updating the initial estimates with a targeted bias-reduction step targeted towards the parameter of interest [13]. The authors of this paper define the data structure for two-stage designs as a missing data structure on the full-data structure X collected in the second phase. The IPCW-TMLE estimator they present is defined as the full-data TMLE with the addition of weights inversely proportional to the probability of being included in the second stage sample (resampled group). In order to use this approach, we would need to define the full data as a right censored data structure (due to administrative censoring) and then define a TMLE capable of addressing right censoring in that full data world. Inverse weighting this TMLE can only recover full efficiency if the full data model is nonparametric. If we have knowledge about the censoring mechanism (for instance), i.e. that censoring is independent or only depends on a subset of covariates, then we cannot recover full efficiency. This estimator still has attractive properties, so we are interested in comparing its performance to that of our proposed estimators.

1.2 Background: Highly Adaptive Lasso

Recall that IPCW estimators which use undersmoothed Highly Adaptive Lasso (HAL) estimators for the weights can improve upon the efficiency of IPCW estimators which use true weights that are known be design [5]. Undersmoothed HAL estimators can also be used to construct asymptotically efficient plug-in estimators [6].

Parametric and semi-parametric approaches to estimating complex regression functions suffer from bias when the specified functional form does not match the truth. Nonparametric methods, on the other hand, provide a highly flexible fit but historically suffer from the "curse of dimensionality." Kernel regression estimators, for instance, are nonparametric but converge at a rate of $O(n^{-k/(2k+d)})$ where d is the dimension of the covariates, k is the degree of the polynomial, and n is sample size. The rate for kernel regression estimators is heavily influenced by the dimension of the covariates (as d increases, the rate will slow down considerably) and strong smoothness assumptions must be made in order to maintain a desirable rate of convergence. This is typical for nonparametric estimators.

In 2016, Benekeser and van der Laan found a way forward when they presented Highly Adaptive Lasso (HAL). Their goal was to construct a nonparametric regression estimator which converges quickly regardless of the dimension of the covariates. To do this, they worked to define a maximum likelihood estimator (MLE) in the class of functions which are right-hand continuous with left-hand limits ("cadlag") and have sectional variation norm bounded by a constant M (a global smoothness constraint). These conditions are considered to be very mild and likely encompass all functions of interest. But why focus on this specific class of functions? Bounded variation cadlag functions form a "Donsker class". Donsker's Theorem, a generalization of the central limit theorem to functions, tells us that functions in a Donsker class converge *quickly* to the true function as sample size grows. Therefore, defining a maximum likelihood estimator (MLE) within the class of bounded variation cadlag functions allows for the construction of a desirable nonparametric regression estimator.

Benkeser and van der Laan discuss a general minimum loss-based estimator in the class of cadlag functions with finite sectional variation norm $\mathbf{\Psi}_M = \{\psi \in \mathbf{\Psi}_M : ||\psi||_v < M\}$. The MLE in this class would be:

$$\psi_{n,M} = argmin_{\psi \in \mathbf{\Psi}_M} \frac{1}{n} \{Y_i - \psi(X_i)\}^2$$

They then present a theorem which shows that this MLE converges to the truth at a rate of $O_P(n^{-1/4 - a(d)/8})$ where $a(d) = 1/(d+1)$. If d is extremely large, the second term in the exponent goes to zero, meaning the worst possible rate of converge for this MLE is $n^{-\frac{1}{4}}$ regardless of the dimension of X.

Next, they leverage the fact that cadlag functions of finite variation norm can be represented as a sum over subsets of an integral with respect to the measure generated on each subset. The measure on each subset can be approximated using discrete support points (points where the step function can jump). This means that the function of interest can be approximately represented as a linear combination of indicator basis functions $x \leftarrow \phi_{s,j}(x)$ with corresponding coefficients $d\psi_{m,s,j}$ summed over s subsets and j support points.

$$\psi_m = \psi(0) + \sum_{s \subset \{1,\ldots,d\}} \sum_{i=1}^{n} I(u_{s,j} \leq x_s) d\psi_{m,s,j}$$

This is a helpful property as we are transforming an optimization problem over all bounded variation cadlag functions into an optimization problem over a finite number of parameters (coefficients of these basis functions). The sum of the absolute value of these coefficients gives the variation norm of the function ψ_M. If we choose the support points to be equal to the actual n observations, then we can define

$$\psi_\beta = \beta_0 + \sum_{s \subset \{1,\ldots,d\}} \sum_{i=1}^{n} \beta_{s,i} \phi_{s,i}$$

and our optimization problem becomes

$$\beta_n = argmin_{\beta, \beta_0 + \sum_{s \subset \{1,\ldots,d\}} \sum_{i=1}^{n} |\beta_{s,i}| < M} P_n L(\psi_\beta).$$

This optimization problem can be solved using Lasso regression under the constraint that the L_1 norm (sum of the absolute value of the coefficients) is bounded by M. This is exactly how HAL is implemented in software. First, the indicator basis matrix is generated and then lasso regression is run on the basis matrix.

HAL is a nonparametric regression estimator and empirical minimizer of the loss function over all cadlag functions of finite sectional variation norm. By employing empirical loss minimization within a Donsker class, HAL achieves an unprecedented rate of convergence, effectively overcoming the "curse of dimensionality" and distinguishing itself from all other machine learning algorithms.

1.3 Background: Undersmoothing HAL

Undersmoothing refers to using a larger L_1 norm than the one selected by cross validation $L_{1,CV}$, which corresponds to using a smaller λ value in HAL's Lasso problem. A smaller λ places a less strict penalty on the magnitude of the coefficients in the model. The "undersmoothed" fit, depending on how you undersmooth, will often include more sparsely supported basis functions than the original fit (meaning basis functions that had coefficients of zero in the original fit may have nonzero coefficients in the undersmoothed fit). Since HAL works to approximately solve the score equations formed by the product of its non-zero basis functions and residuals, including a larger number of basis functions in the fit results in solving more score equations. The canonical gradient is a score, so increasing the number of scores solved by HAL increases the likelihood that the linear span of the score equations solved by HAL can begin to approximate the score equation implied by the canonical gradient.

It is often desirable to produce an undersmoothed HAL fit that includes a larger number of nonzero basis functions relative to an initial HAL fit. One simple way to do this is to fit an initial HAL model with λ_{CV} and then refit the model on a series of smaller λs (larger L_1 norms). The λ which gives the best performance can then be selected for the final fit. The difficulty with this approach can be choosing an optimal λ – Lepski's method can be used but requires getting an estimate of the variance for each estimator with each different value of λ.

Another estimation procedure which results in an undersmoothed HAL fit is called "relaxed HAL." This procedure first fits an initial HAL model with L_1-norm chosen by cross-validation. The nonzero basis functions from the initial fit are then refit in a model without penalization, which produces a fit with a larger L_1 norm (this second step is akin to running glm() in R on the nonzero basis functions from the initial HAL fit). This estimation procedure works to solve the same score equations as the initial HAL fit but solves them exactly (by removing the regularization). This method will never solve more score equations than the original HAL fit, a weakness relative to other approaches. It is also an aggressive undersmoother and may lead to overfitting in some cases. However, this method is attractive as it is both easier to implement in practice and less computationally expensive than the first method described.

Using relaxed HAL can cause too large of an increase in L_1 norm relative to $L_{1,CV}$, which may indicate an overfitting problem. If this occurs, one can instead fit an initial cross-validated HAL model and then iteratively increase the L_1 until the score equations are solved at a rate of $\frac{\sigma_n}{\sqrt{n log(n)}}$ [14]. Like relaxed HAL, this method does not increase the number of score equations solved but does produce an undersmoothed fit.

1.4 Background: RAL Estimators & Efficiency Theory

The goal of this paper is to identify the most efficient estimator for our target parameter. This search is constrained to the class of regular and asymptotically linear (RAL) estimators, the class of "well-behaved" estimators. To provide intuition for this restriction, we will rely on discussions from [15]. Many of the results concerning efficiency and the behavior of RAL estimators, however, were established much earlier in [16].

An estimator is asymptotically linear if it has influence curve ϕ where $\mathbb{E}[\phi(Z)] = 0$ and satisfies:

$$\sqrt{n}\left[(\hat{\psi}_n - \psi) - P_n\phi(Z)\right] \rightarrow^p 0$$

As n gets big, the difference between an asymptotically linear estimator and the truth behaves like the average of iid random variables. Given the central limit theorem, we can say that the sampling distribution of an asymptotically linear estimator converges to a normal distribution with mean 0 as n increases. The asymptotic variance of the estimator is determined by the variance of the estimator's influence curve. The influence curve is a mathematical object which tells us how much each observation influences the resulting estimate and is useful in quantifying the variance of our estimator.

$$\sqrt{n}(\hat{\psi}_n - \psi) \rightarrow^d N(0, V[\phi(Z)])$$

If we use a RAL estimator, we automatically have at least one way of constructing asymptotic confidence intervals – through use of the influence curve. However, RAL estimators are *asymptotically* linear, so we must be aware of how much finite sample bias comes from the "second-order" remainder $R_n(P_n, P) = [\hat{\psi}_n(P_n) - \psi(P)] - P_n\phi(Z)$.

Regularity has been described "robustness to distributional shifts" [15]. Regular estimators still converge to the same normal distribution asymptotically if instead of drawing data from the true distribution P we draw from any sequence of distributions that goes towards P quickly enough. A previous work provides a great example which demonstrates the concept of regularity well [15]. If we proposed an estimator for the population mean defined by $\hat{\psi}(P_n) = 5$, it would work perfectly if the true population mean were indeed 5 and is even asymptotically linear! However, if the true population mean were anything else, this estimator would perform terribly. Regularity would rule this estimator out by making sure that the estimator attains the same limiting distribution under a shifted version of the true distribution as it does under the true distribution itself.

All RAL estimators have influence curves, and the variance of a RAL estimator is determined by the variance of its influence curve. Thus, the best RAL estimator will be the one whose influence curve has the smallest variance. This smallest-variance influence curve is called the "efficient influence curve." Influence curves are estimator-specific, but the efficient influence curve is parameter-specific. This means that we can derive the efficient influence curve for our target parameter and statistical model and then work on constructing an estimator which has influence curve equal to the efficient influence curve.

If our target parameter is pathwise differentiable, there is only one efficient influence curve, and it lies in the tangent space of our statistical model. We define tangent spaces in Chapter 3 and discuss how to mathematically derive the efficient influence curve for our problem – a key step in constructing efficient estimators.

Chapter 2

Methodology

This section will follow the “roadmap” for statistical learning [7]. The roadmap is a formal framework created to standardize the approach to answering questions in causal inference.

2.1 Defining the Estimation Problem

Data

Let t=0 be the baseline time that a patient is seen in a clinic for the first time where $t = 0, ..., \tau$ for max follow up time τ, which can vary across patients. Let $X_1(0)$ be baseline covariates, $D(t) = I(T \leq t)$ be a possibly unobserved indicator of death by time t (with T denoting the time at which $D(t)$ jumps to 1), $V(t)$ be an indicator of attending a clinic visit at time t, $A_2(t)$ be an indicator of being administratively right censored by time t, $Id(t)$ be an indicator of death being reported by time t prior to τ, and $L(t)$ be any time-dependent covariates collected at clinic visits (note $L(t)$ can only be measured at t if $V(t) = 1$ at that time. To simplify notation going forward, let $X_1(t) = (Id(t), V(t), L(t)V(t))$.

Full Data

Suppose we are interested in understanding cumulative risk of mortality at a specified time t_0. The “full data” X represents the data we would like to have observed in an ideal world. We choose to define the full data as having no right censoring by study end before t_0 and outcomes measured on all participants at t_0. In notation, the full data can be written $X = (X_1(0), \bar{X}(t_0), \bar{D}(t_0)) \sim P_X$. Overbar notation will be used to denote the entire history of the variable up until the time shown in parenthesis. Note that X includes baseline covariates, the history of clinic visits until t_0, and the true death status up until the time of interest t_0. In the full data, $\tau \geq t_0$ for each person. This indicates that all patients are followed up at least until the time of interest for mortality estimation in the full data world.

Observed Data (after resampling)

In reality, we do not get to observe the full data due to incomplete death reporting and right-censoring by study endpoint τ. “Resampling”, or extensive tracking to ascertain the vital status of a subset of patients with unknown outcome status, can help recover some of this unobserved data. Let Tr be an indicator that a person is chosen for resampling or that their vital status is already known. As researchers, we set the resampling probabilities $P(Tr(t) = 1 \mid X_1(0), \bar{X}(t), \bar{D}(t)^{Tr(t)}, \bar{Tr}(t-1), \bar{A}_2(t-1))$ and have control over who is more likely to be sampled in the second stage. We observe each subject until τ, so our final observed data is given by:

$$O = \bar{O}(\tau) = (X_1(t), A_2(t), Tr(t), Tr(t)\bar{D}(t+1) : t \leq \tau)$$

Note that we have $\bar{D}(t+1)$ since death is recorded one additional time, after resampling.

We choose to put a few simplifying constraints on the problem for the time being. Note, these assumptions will not necessarily hold in practice and future work will address these gaps.

- Constraint 1: Resampling occurs only once, at the end of each patient's study period τ (if a patient is selected).
- Constraint 2: We successfully track everyone that is chosen for resampling (ie a true vital status is ascertained for everyone chosen for resampling).

We can view the time ordering of the data up to some max time τ as follows where $D(\tau)$ represents an updated vital status post-resampling. Recall τ can vary across individuals.

$$\begin{aligned} O = (&X_1(0), A_2(0), Id(1), D(1)^{Tr(t)}, V(1), V(1)Lt(1), Tr(1), A_2(1), ..., \\ &V(t), V(t)Lt(t), Tr(t), A(t), I(\text{resampled}), Tr(\tau), D(\tau)^{Tr(\tau)}) \end{aligned}$$

Statistical Model

Now we will define the statistical model, which represents all possible distributions of the observed data [13]. This model should reflect our real world knowledge of the data generating distribution.

Factorizing the Likelihood

The likelihood of the observed data can be represented in its factorized form as follows.

$$\begin{aligned} p(O) \quad = \quad & X_1(0) \prod_{t \leq \tau} p(X_1(t) \mid X_1(0), \bar{X}(t-1), \bar{A}_2(t-1)) \times \\ & \prod_{t \leq \tau+1} p(\bar{D}(t) \mid X_1(0), \bar{X}(t), \bar{A}_2(t-1), Tr(t) = 1)^{Tr(t)} \times \\ & \prod_{t < \tau} p(Tr(t) \mid X_1(0), \bar{X}(t), \bar{D}(t)^{Tr(t)}, \bar{Tr}(t-1), \bar{A}_2(t-1)) \times \\ & p(Tr(\tau) \mid X_1(0), \bar{X}(\tau), \bar{D}(\tau)^{Tr(\tau)}, \bar{Tr}(\tau-1), \bar{A}_2(\tau-1)) \times \end{aligned}$$

$$\prod_{t\leq\tau} p(A_2(t)|X_1(0), X_1(t), \bar{D}(t), Tr(t))$$

Note that $X_1(t) = (Id(t), V(t), V(t)L(t))$, so we can factorize this density further; this more extensive factorization will be presented in the section on Estimator 3. Let $g_{A_2(t)} = P(A_2(t)|Pa(A_2(t)))$ for $t \leq t_0$ and let $g_{Tr(\tau)} = P(Tr(\tau)|Pa(Tr(\tau)))$; these are the intervention nodes or “g” factors of our likelihood. We use Pa notation to denote the “parents” of a variable or all variables that precede said variable in time order and affect its realized value.

Defining the Statistical Model

Our statistical model $\mathcal{M}^F$ is defined by knowing the probability of having known status $p(Tr(t) \mid \bar{X}(t), X_1(0), \bar{A}_2(t-1))$ and the probability of being right censored by study endpoint $P(A_2(t)|X_1(0), X_1(t), \bar{D}(t)^{Tr(t)}, Tr(t))$ by design and leaving all other factors of the density of $P(O)$ completely unspecified. These restrictions reflect reality as we set the resampling probabilities ourselves and know the participants' max follow-up times τ as baseline.

Structural Equations

In order to define structural equations, it is useful to rewrite the data in longitudinal form as follows:

$$O = (K(0), A(0), ..., K(\tau), A(\tau), Y = K(\tau + 1))$$

In this time ordered representation, $A(t)$ is our intervention node. For time points $t = 0, ..., \tau-1$, the intervention node only includes the right censoring indicator which corresponds to study endpoint $A(t) = A_2(t)$. However, at time $t = \tau$, we have two interventions, so the node represents both right censoring and resampling $A(\tau) = (A_2(\tau), Tr(\tau))$. $K(0)$ is our baseline covariate. $K(t) = (X_1(t), D(t)^{Tr(t)}, Tr(t))$ represents the variables we track through time. $K(\tau + 1) = Y$ is our outcome variable, which is realized after τ and resampling. From here, consider the nonparametric structural equation model (NPSEM) defined below:

$$\begin{aligned} K(t) &= f_{K(t)}\left(\bar{A}(t-1), \bar{L}(t-1), U_{K(t)}\right), t = 0, ..., \tau + 1 \\ A(t) &= f_{A(t)}\left(\bar{A}(t-1), \bar{L}(t), U_{A(t)}\right), t = 0, ..., \tau \end{aligned}$$

Our interventions of interest are then to set $A_2(t) = 0$, for all $t < t_0$, and $Tr(\tau) = 1$ if vital status is not already known (this is equivalent to resampling all individuals with unknown outcome status at their study endpoint). The post intervention distribution replaces the structural equations in the SCM for $A_2(t)$ and $Tr(\tau)$ with deterministic functions that set all of the right censoring (by study end) nodes to uncensored (up until t_0) and the resampling node to 1 for all patients with unknown outcomes.

Target Parameter

A target parameter is a function of the full data X that represents the mathematical quantity that we are interested in learning. More rigorously, the target parameter can be defined as a function Ψ of P_X that maps the true probability distribution of our data to the target feature of interest [13]. Recall, our goal is to estimate a mortality rate among patients receiving care for HIV. Our target parameter is $\Psi(P_X) = P(T \leq t_0)$. In words, we are interested in knowing the probability of death happening before or at a specified time of interest, t_0, among patients in our population of interest in the full data world where all participants have observed outcomes and no one is censored.

Identification

The target parameter is next rewritten as a function of the observed data through a process known as "identification". Our target parameter can be viewed as a function of the observed data by defining it as an expectation of $Y(t_0) = I(T \leq t_0)I(\tau \geq t_0, Tr(\tau) = 1)$ under the G-computation density which replaces p_{A_2} with a density that guarantees $A_2(t) = 0$ (no right censoring by study end) for all $t = 0, ..., t_0$ and replaces $p(Tr(\tau) \mid X_1(0), \bar{X}(\tau), \bar{D}(\tau)^{Tr(\tau)}, \bar{Tr}(\tau-1), \bar{A}_2(\tau-1))$ with $I(Tr(\tau) = 1)$. By sequential randomization of τ and $Tr(\tau)$, $E[I(T \leq t_0)I(\tau \geq t_0, Tr(\tau) = 1] = P(T \leq t_0)$. Then, our target parameter would be identified as an expectation of $Y(t_0)$ under the G-computation density.
G-computation Density

$$\begin{aligned}
P^{g^*}(o) \quad = \quad & \prod_{t \leq \tau} p(X_1(t) \mid X_1(0), \bar{X}(t-1), \bar{A}_2(t-1)) \times \\
& \prod_{t \leq \tau+1} p(\bar{D}(t) \mid X_1(0), \bar{X}(t), \bar{A}_2(t-1), Tr(t) = 1)^{Tr(t)} \times \\
& \prod_{t < \tau} p(Tr(t) \mid X_1(0), \bar{X}(t), \bar{D}(t)^{Tr(t)}, \bar{Tr}(t-1), \bar{A}_2(t-1)) \times I(Tr(\tau) = 1) \times \\
& \prod_{t \leq t_0} I(A_2(t) = 0) \prod_{t_0 < t \leq \tau} p(A_2(t) | X_1(0), X_1(t), \bar{D}(t)^{Tr(t)}, Tr(t))
\end{aligned}$$

Assumptions
In order to identify our target parameter as an expectation under the G-computation density, we need to make a few standard assumptions, as mentioned above. Recall once again that our intervention node $A(t)$ represents the censoring indicator at all time points but also represents the resampling node when $t = \tau$.

1. $Y(t_0) \perp A_2(t) \mid \bar{K}(t), \bar{A}(t-1)$ for all $t = 0, ..., \tau$
2. $Y(t_0) \perp Tr(\tau) \mid \bar{K}(t), \bar{A}(t-1)$ for all $t = \tau$
3. $P(A_2(t) = 0 \mid \bar{K}(t), \bar{A}(t-1) = \bar{a}(t-1)) > 0$
4. $P(Tr(\tau) = 1 \mid \bar{K}(t), \bar{A}(t-1) = \bar{a}(t-1)) > 0$

Assumptions (1) and (2) are the sequential randomization assumptions (SRA) which assume that each intervention node is independent of the counterfactual outcome $Y(t_0)$ given the observed past at all relevant t. These assumptions will be met since our "interventions" do not affect T directly but rather determine whether or not T is observed. Assumptions (3) and (4) are the sequential positivity assumptions which assume a positive probability of right censoring (by study end) throughout the study and a positive probability of resampling at the end of the study in all relevant covariate strata. Under these two assumptions, our target parameter can be written as a function of the observed data through the longitudinal G-computation formula.

Calculating the Efficient Influence Curve

Now the statistical estimation problem has been fully defined, and we can move on to constructing an ideal estimator, guided by efficiency theory. The next step is to derive the efficient influence curve for the statistical estimand of interest in our statistical model. One method for deriving the efficient influence curve involves finding the influence curve of any RAL estimator of our target parameter and projecting it onto the tangent space of the statistical model. We describe this process in detail in the following sections.

Tangent Space of the Model

The tangent space of a model is a subspace of $L_0^2(P)$ defined by the linear span of the set of all scores of submodels through P (note that the tangent space only depends on the statistical model and the true distribution P, not the estimator or parameter) [13]. Tangent spaces can be represented as orthogonal sums of multiple sub-tangent spaces if the model has a variation independent parameterization. The tangent space of our model $T(P_X)$ is equal to $T_{t,X_1} \oplus T_{t,T}$ across time $t \leq \tau$. Recall, once again, that $X_1(t)$ represents a collection of variables.

Initial Gradient

Inverse probability of censoring weighted (IPCW) estimators are RAL estimators, so their influence curves can be used as initial gradients in the process of deriving the efficient influence curve. An IPCW estimator is derived below for this purpose.

IPCW Function

There are two interventions interest. First, resample everyone with unknown outcome status at study end point by setting $Tr(\tau) = 1$ if $Tr(\tau) = 0$ otherwise. Second, set right censoring (by study end) $A_2(t) = 0$) at all time points up until t_0 (equivalently set $\tau \geq t_0$ where τ is time of right censoring). In order to observe $I(T \leq t_0)$, we need both of these conditions to hold. The IPCW function for our problem can be represented as:

$$\frac{I(T \leq t_0) I(\tau \geq t_0, Tr(\tau) = 1)}{P(\tau \geq t_0, Tr(\tau) = 1 \mid X)}$$

Next, we must find an expression that will allow us to estimate the probabilities in the denominator. This is where our IPCW function deviates slightly from those typically used. One of our interventions enforces that $\tau \geq t_0$. This intervention is complex in the sense that $\tau \geq t_0$ does not correspond to one particular intervention (i.e. enforcing $\tau = 3$). There are many possible interventions that satisfy this criteria. For example, if $t_0 = 3$, we could intervene to set $\tau = 3$ or $\tau = 4$ or $\tau = 5$. Therefore, we must integrate over all possible values of τ that satisfy this criteria. Since τ only takes discrete values, we can express this quantity as a sum in the end. The full derivation is presented below (Note: $G(ds \mid X) = G_n(ds))$):

$$\begin{aligned}
P(\tau \geq t_0, Tr(\tau) = 1 \mid X) &= \int_{s \geq t_0} P(\tau \in ds, Tr(s) = 1 | X) \\
&= \int_{s \geq t_0} P(Tr(s) = 1 \mid \tau = s, X) P(\tau \in ds | X) \\
&= \int_{[t_0, \infty)} P(Tr(s) = 1 \mid \tau = s, X) G(ds \mid X) \\
&= \int_{[t_0, \infty)} P(Tr(s) = 1 \mid \tau = s, X) G_n(ds) \\
&= \sum_{s \geq t_0} P_\tau(s \mid X) P(Tr(s) = 1 \mid \tau = s, X)
\end{aligned}$$

Here, $P_\tau(s \mid X) = \prod_{s \leq t}(1 - \lambda(s \mid X))\lambda(t \mid X)$ can be defined in terms of the hazard $\lambda(t \mid X)$ of being censored at time t. Our final IPCW function is:

$$\frac{I(T \leq t_0) I(\tau \geq t_0, Tr(\tau) = 1)}{\sum_{s \geq t_0} P_\tau(s \mid X) P(Tr(s) = 1 \mid \tau = s, X)}$$

In expectation, this function equals $\Psi(P_X)$. For notational convenience, let $\Pi_\tau(X) = \sum_{s \geq t_0} P_\tau(s \mid X) P(Tr(s) = 1 \mid \tau = s, X)$. An empirical mean of $I(T \leq t_0) I(\tau \geq t_0, Tr(\tau) = 1)/\Pi_\tau(P_X)$ is an unbiased linear estimator of $\Psi(P_X)$, so its influence curve is an initial gradient of the pathwise derivative of Ψ. For the purpose of finding the canonical gradient, we can use it as an initial influence curve/gradient:

$$D = \frac{I(T \leq t_0) I(\tau \geq t_0, Tr(\tau) = 1)}{\Pi_\tau(X)} - \Psi(P_X)$$

Cannonical Gradient

The cannonical gradient can be found by projecting the initial gradient, D found above, onto the tangent space $T(P)$ of the model $\mathcal{M}$. Since the tangent space can be represented as an orthogonal sum of tangent spaces $T_{t,X_1} + T_{t,T}$, projecting D onto each orthogonal tangent space and summing will result in the projection onto the entire tangent space and thus the efficient influence curve.

The general form of a projection of an initial gradient D onto a factor of the tangent space corresponding to the variable L is given by:

$$\mathbb{E}(D \mid L, Pa(L)) - \mathbb{E}(D \mid Pa(L))$$

Recall that we must compute this projection for each of our factorized tangent spaces. We then orthogonally sum these projections to obtain the efficient influence curve:

$$
\begin{aligned}
&D^* = \Pi(D \mid T(P_X)) \\
&= \textstyle\sum_t I(t \leq \tau) E\left(\frac{I(T \leq t_0) I(\bar{A}_2(min(t,t_0))=0, Tr(\tau)=1)}{\Pi_\tau(X)} \mid X_1(t), \bar{X}(t-), \bar{A}_2(t)\right) \\
&- \textstyle\sum_t I(t \leq \tau) E\left(\frac{I(T \leq t_0) I(\bar{A}_2(min(t,t_0))=0, Tr(\tau)=1)}{\Pi_\tau(X)} \mid \bar{X}(t-), \bar{A}_2(t)\right) \\
&+ \textstyle\sum_t I(t \leq \tau) I(Tr(t) = 1) \\
&E\left(\frac{I(T \leq t_0) I(\bar{A}_2(min(t,t_0))=0, Tr(\tau)=1)}{\Pi_\tau(X)} \mid \bar{D}(t), \bar{X}(t-), X_1(t), \bar{A}_2(t), Tr(t) = 1\right) \\
&- \textstyle\sum_t I(t \leq \tau) I(Tr(t) = 1) \\
&E\left(\frac{I(T \leq t_0) I(\bar{A}_2(min(t,t_0))=0, Tr(\tau)=1)}{\Pi_\tau(X)} \mid \bar{X}(t-), X_1(t), \bar{A}_2(t), Tr(t) = 1\right) \\
&= \textstyle\sum_t I(t \leq \tau)\{P(T \leq t_0 \mid X_1(t), \bar{X}(t-)) - P(T \leq t_0 \mid \bar{X}(t-))\} \\
&+ \textstyle\sum_t I(t \leq \tau) I(Tr(t) = 1)\{P(T \leq t_0 \mid \bar{D}(t), X_1(t), \bar{X}(t-)) - P(T \leq t_0 \mid X_1(t), \bar{X}(t-))\}
\end{aligned}
$$

Note that this eIC includes contributions from people for whom $\tau \leq t_0$ on all points up until they are censored as we are summing over all t. This is a promising sign for efficiency when building estimators with influence curve equal to the efficient influence curve.

2.2 Estimator 1: IPCW

The first estimator we examine is an inverse probability of censoring weighted (IPCW) estimator (this estimator was derived in detail for this application previously in the "initial gradient" section). IPW estimators are regular and asymptotically linear (RAL), and they are defined as a solution to an estimating equation. These estimators rely on specifying models for all intervention mechanisms. Estimating these propensity scores helps to identify subjects who are underrepresented in the data and upweight them to construct a pseudo-population in which selection biases are eliminated. Our IPCW estimator is:

$$\frac{1}{n}\sum_{i=1}^{n}\left[\frac{I(T \leq t_0) I(\tau \geq t_0, Tr(\tau) = 1)}{\sum_{s \geq t_0} P_\tau(s \mid X) P(Tr(s) = 1 \mid \tau = s, X)}\right] = \mathbb{E} I(T \leq t_0) = \Psi(P_X)$$

IPW estimators are asymptotically efficient, but we suspect this estimator may be substantially inefficient in finite samples as it relies on stratification of the data by $\tau \geq t_0$. It nonetheless serves as a useful baseline metric to compare with the HAL-based estimators presented in the next sections.

2.3 Estimator 2: IPCW-HAL

It has recently been shown that the efficiency of IPW estimators can be improved when undersmoothed Highly Adaptive Lasso (HAL) estimators are used to estimate the propensity scores [5]. This holds even in our case, where the true treatment probabilities are known by design.

IPW estimators are relatively straightforward to construct and implement in practice, making them a popular choice for estimating causal effects. However, they require correct specification of the models for the weights, and their rate of convergence depends upon the rate of convergence of the proposed models for those weights [5]. While parametric models for the propensity scores provide a sufficient rate of convergence, they are difficult to correctly specify given our limited knowledge of the true data generating distribution and may lead to inconsistent estimates of the target parameter if misspecified. IPW estimators are also known to be inefficient in a variety of settings when compared to alternatives.

Use of flexible machine learning algorithms can help to avoid misspecifying the model for the weights, but these algorithms often suffer from insufficient rates of convergence in high dimensional data. However, Highly Adaptive Lasso (HAL) estimators provide both flexibility and a guaranteed fast rate of convergence regardless of the dimension of the covariates, making them a sensible choice for modeling weights for IPW estimators [17][5].

We present an IPCW-HAL estimator, which is defined as a projection of the IPCW *function* D (not the IC of the IPCW estimator as defined previously) onto the tangent space of our intervention mechanism G. The influence curve of this IPCW-HAL estimator is defined as $D^*(P) = D - D_{CAR}(P)$ where D_{CAR} is the projection of the IPCW function onto T_{CAR}. Under coarsening at random, let T_{CAR} be the tangent space which corresponds to our intervention mechanisms $A_2(t)$ and $Tr(\tau)$. The projection $D_{CAR}(P) = \Pi\{D \mid T_{CAR}\}$ can be broken down a sum of the projections of D onto $T_{Tr(\tau)}$ and $T_{A_2(t)}$:

$$\begin{aligned}
\Pi\{D \mid T_{Tr(\tau)}\} &= I(Tr(\tau) = 0)\,[E(D \mid Tr(\tau), Pa(Tr(\tau))) - E(D \mid Pa(Tr(\tau)))] \\
&= I(Tr(\tau) = 0)\,[E(D \mid Tr(\tau) = 1, Pa(Tr(\tau))) \\
&\quad - E(D \mid Tr(\tau) = 0, Pa(Tr(\tau))]) \\
&\quad \times (Tr(\tau) - P(Tr(\tau) = 1 \mid Pa(Tr(\tau))]))
\end{aligned}$$

$$\begin{aligned}
\Pi\{D \mid T_{A_2(t)}\} &= \sum_t I(A_2(t-) = 0)\,[E(D \mid A_2(t), Pa(A_2(t))) - E(D \mid Pa(A_2(t)))] \\
&= \sum_t I(A_2(t-) = 0)\,[E(D \mid A_2(t) = 1, Pa(A_2(t))) \\
&\quad -E(D \mid A_2(t) = 0, Pa(A_2(t)))] \\
&\quad \times (A_2(t) - P(A_2(t) = 1 \mid Pa(A_2(t)))))]
\end{aligned}$$

$$D_{CAR}(P) = \Pi\{D \mid T_{Tr(\tau)}\} \oplus \Pi\{D \mid T_{A_2(t)}\}$$

$$D^*(P) = D - D_{CAR}(P)$$

If the full data model were nonparametric, the influence curve of this IPCW-HAL estimator would be equivalent to the eIC, meaning the IPCW-HAL estimator would be fully efficient [6]. Unfortunately, this will not hold in our application. In causal inference problems, we typically believe that there are different counterfactual outcomes under different interventions (that the intervention truly changes the outcome). Here, patients do not have a different counterfactual outcome depending on whether or not they were censored; the true outcome is still the same under many possible interventions. This means we have more knowledge about the structure of our model than in typical causal inference problems. Our full data model is not completely nonparametric. The influence curve of our IPCW-HAL estimator will not be equal to the eIC. While we can gain efficiency by using undersmoothed HAL to estimate weights that are known by design, we will never be fully efficient with this strategy. This realization provides motivation for the construction of Estimator 3, which provides an avenue for constructing a fully efficient estimator.

In order to implement the IPCW-HAL estimator, will use undersmoothed HAL estimators to estimate the propensity scores shown in the denominator of our IPCW function:

$$\sum_{s \geq t_0} P_\tau(s \mid X) P(Tr(s) = 1 \mid \tau = s, X)$$
$$P_\tau(s \mid X) = \prod_{s \leq t} (1 - \lambda(s \mid X)) \lambda(t \mid X)$$

The hazard of right censoring by study endpoint $\lambda(t \mid X)$ will need to be estimated at each t. Using information from the covariate history up until t can help with efficiency gains. $P(Tr(s) = 1 \mid \tau = s, X)$ will also need to be estimated for every $\tau \geq t_0$ in the same manner.

The amount of undersmoothing should be chosen such that the basis functions in the HAL fits of the intervention mechanisms are sufficient to approximate the outcome regression function within an $n^{-1/4}$ neighborhood [5]. This may hold without undersmoothing if G and f have similar complexity. More undersmoothing is likely required when G is less complex than f (the outcome regression function) because the initial HAL fit for G (without undersmoothing) will not generate a sufficiently large set of basis functions to explain the variation in the more complex f [5]. We also want to include a rich enough set of basis function to ensure that the efficient influence curve equation is solved at a sufficient rate.

2.4 Estimator 3: Plug-in Estimator Based on Undersmoothed HAL

Plug-in estimators require estimation of the observed data conditional densities in the likelihood factorization. These estimates are then used to estimate the target parameter by (1) integrating with respect to the estimated distribution of the covariates under the specified interventions or (2) Monte Carlo simulation. Once counterfactual data has been simulated, evaluation of the the target parameter of interest in this problem amounts to taking an empirical mean of the indicator of death at t_0.

In 2021, van der Laan, Benkeser, and Cai presented results for the efficient estimation of pathwise differentiable target parameters using the undersmoothed Highly Adaptive Lasso estimator [6]. Their paper presents three properties which dictate the efficiency of a plug-in estimator:

1. Negligibility of the empirical mean of the canonical gradient
2. Control of the second-order remainder
3. Asymptotic equicontinuity

They go on to explain how undersmoothed HAL estimators can help us achieve all three desired properties and lead to efficient plug-in estimators for pathwise differentiable target parameters. Briefly, we will discuss each of these three conditions.

Negligibility of the empirical mean of the canonical gradient can be achieved by undersmoothing the HAL fit enough so that the generated bases are rich enough to ensure that the efficient score equation is solved up to an appropriate level of approximation. Since undersmoothing leads to the inclusion of additional basis functions in the resulting fit, it is a strategy geared towards this goal.

For a plug-in estimator to be efficient, we also need the second order remainder to be $o_P(n^{-1/2})$. We need to preserve the $n^{-1/4}$ rate of convergence that HAL achieves when the L_1 norm bound is selected with cross-validation. Results from van der Laan, Benkeser, and Cai state that the HAL's rate of convergence will not be affected by undersmoothing so long as we monitor the L_1 norm bound to ensure that it does not exceed the sectional variation norm of the function of interest.

Lastly, results from emprical process theory tell us that asymptotic equicontinuity holds for empirical processes indexed by a Donsker class. Since the class of cadlag functions with bounded sectional variation norm is an example of a Donsker class and HAL assumes the function of interest to be of this type, we satisfy the third condition so long as we are not violating the assumptions inherent to establishing the HAL's rate in the first place.

With these three key ingredients in hand, we are in a good position to construct an efficient plug-in estimator. In order to do so, we will have to estimate each non-intervention factor of our likelihood using undersmoothed HAL estimators. Recall our previous factorization of the likelihood but now with the $X_1(t)$ expanded out fully (since it contains several variables):

$$\begin{aligned} p(O) \quad = \quad & X_1(0)\prod_{t\leq\tau} p(Id(t) \mid X_1(0), \bar{X}(t-1), \bar{A}_2(t-1)) \times \\ & p(V(t) \mid X_1(0), \bar{X}(t-1), \bar{A}_2(t-1)) \times \end{aligned}$$

$$
\begin{aligned}
&p(Lt(t) \mid X_1(0), \bar{X}(t-1), \bar{A}_2(t-1)) \times \\
&\prod_{t \leq \tau+1} p(\bar{D}(t) \mid X_1(0), \bar{X}(t), \bar{A}_2(t-1), Tr(t) = 1)^{Tr(t)} \times \\
&\prod_{t < \tau} p(Tr(t) \mid X_1(0), \bar{X}(t), \bar{D}(t)^{Tr(t)}, \bar{Tr}(t-1), \bar{A}_2(t-1)) \times \\
&p(Tr(\tau) \mid X_1(0), \bar{X}(\tau), \bar{D}(\tau)^{Tr(\tau)}, \bar{Tr}(\tau-1), \bar{A}_2(\tau-1)) \times \\
&\prod_{t \leq \tau} p(A_2(t) | X_1(0), X_1(t), \bar{D}(t)^{Tr(t)}, Tr(t))
\end{aligned}
$$

Chapter 3

Simulations & Results

Simulated data used here mimics summary statistics from previous resampling-based studies which followed individuals on antiretroviral therapy (ART) over time [1]. However, one continuous covariate representing a CD4 count was binarized for the purposes of this thesis, and the mortality and right censoring (by end of study) rates were inflated in the simulated data to avoid small effective samples sizes (which can be problematic when fitting HAL models). We leave these added complications to future work. Details of the data generating code can be found in the supplementary material.

IPCW and IPCW+HAL estimators were implemented on simulated data of size $n = 100, 1000, 10000$ (Note: we did not apply IPCW+HAL estimators to data of N=100 due to algorithmic constrains). Section 3.1 presents results from simulation studies done in the case where resampling is done completely randomly among patients with unknown vital status at their study endpoint. Section 3.2 presents results from simulation studies done in the case where resampling probabilities depend on the patients' covariate history. In simulation two, patients have a higher probability of being resampled if they miss a larger number of visits. These "Dependent Resampling" simulations were run mostly as a check to ensure that our estimators function properly in preparation for future work which sets resampling probabilities in a data-adaptive manner to optimize efficiency gains.

Whenever HAL models were fit for the IPCW-HAL estimators, zero order splines were used (this was the only non-default argument specified). Default settings were used for all other arguments to "fit_hal" in the "hal_9001" R package. Most notably, the highest degree of interaction used to generate the basis functions was set to two given the large number of predictors. Undersmoothing, when used, was done by refitting the HAL model on a series of $\lambda < \lambda_{CV}$ as seen in the results tables below. This method was chosen over "relaxed HAL" since the initial HAL fits for the right censoring mechanisms (by study endpoint) returned intercept only models (which made sense given the simple, independent manner in which right censoring time was generated). Recall that relaxed HAL refits the non-zero coefficient basis functions from an initial HAL fit (which uses λ_{CV}) but without regularization. Neither HAL nor lasso regression, as implemented in R, penalize the intercept term to begin with. Thus, relaxed HAL accomplishes nothing when one begins with an intercept-only model. With this in mind and the desirable nature of undersmoothing in a manner that includes sparsely supported basis functions in the updated hal fit, we choose our method of undersmoothing over relaxed HAL.

3.1 Simulation Results for IPCW vs IPCW + HAL Estimators

Simulation 1: Independent Resampling

Simulation study 1 shows a 32% decrease in variance between the IPCW and the "best" IPCW-HAL estimator on data of size N=1k. Similarly, we see a 38% decrease in variance for sample size N=10k. Though we have not discussed how to determine the optimal degree of undersmoothing for IPCW-HAL estimators, for now take this to be the estimator with the smallest variance among choices of λ (in practice, the resulting change in bias relative to the change in standard error should be considered as well). Note the finite sample bias seen in these tables may largely come the Monte Carlo simulation done to compute the "truth" ψ_0. Future work will look into more precise calculations of ψ_0 or changing simulation settings such that this error is less relevant. This applies to all simulations in this thesis.

Table 3.1 displays the performance of the IPCW estimator which uses known probabilities for the weights across all three sample sizes. The variance of this estimator decreases, as expected, with increased sample size.

	N	true ψ_0	Mean Est	Bias	Var	MSE	SD	t0
1	100	0.2162	0.2177	0.0019400	0.0034736	0.0034774	0.0589372	4
2	1000	0.2162	0.2177	0.0014808	0.0003407	0.0003429	0.0184594	4
3	10000	0.2162	0.2177	0.0015599	0.0000349	0.0000373	0.0059077	4

Table 3.1: Simulation results for IPCW estimator which uses known probabilities for weights (Independent Resampling). These results were based on 10k simulations of independently drawn data.

Table 3.2 displays results for our IPCW-HAL estimator fit on data of sample size N=1k. The first row shows the performance for an IPCW-HAL estimator which uses $\lambda = \lambda_{CV}$. The IPCW-HAL estimator corresponding to row one of the table is not undersmoothed. The subsequent rows show these same results for undersmoothed IPCW-HAL estimators with the amount of undersmoothing increasing as the row index increases. Row 10 shows the IPCW-HAL estimator with the most amount of undersmoothing. The estimator in row 10 also has the smallest variance. Note that the bias also decreases as we undersmooth more until row 9 where the bias starts to increase once again. This is the phenomenon we must pay attention to when discussing the "optimal" amount of undersmoothing. For now, we refer to the IPCW-HAL estimator with $\lambda = 0.1 * \lambda_{CV}$ as the "best" estimator for this sample size.

	N	λ	true ψ_0	Mean Est	Bias	Var	MSE	SD	t0
1	1000	λ_{CV}	0.2160	0.2165	0.0004210	0.0003106	0.0003108	0.01762	4
2	1000	0.9*λ_{CV}	0.2160	0.2164	0.0003890	0.0003062	0.0003064	0.01750	4
3	1000	0.8*λ_{CV}	0.2160	0.2164	0.0003661	0.0003017	0.0003018	0.01737	4
4	1000	0.7*λ_{CV}	0.2160	0.2164	0.0003536	0.0002967	0.0002968	0.01722	4
5	1000	0.6*λ_{CV}	0.2160	0.2164	0.0003288	0.0002913	0.0002915	0.01707	4
6	1000	0.5*λ_{CV}	0.2160	0.2163	0.0002811	0.0002858	0.0002859	0.01691	4
7	1000	0.4*λ_{CV}	0.2160	0.2162	0.0001866	0.0002802	0.0002802	0.01674	4
8	1000	0.3*λ_{CV}	0.2160	0.2161	0.0000083	0.0002742	0.0002742	0.01656	4
9	1000	0.2*λ_{CV}	0.2160	0.2158	-0.0002573	0.0002677	0.0002677	0.01636	4
10	1000	0.1*λ_{CV}	0.2160	0.2154	-0.0006925	0.0002577	0.0002582	0.01605	4

Table 3.2: Simulation results for IPCW-HAL estimators across different levels of undersmoothing applies to data sets of size 1k (Independent Resampling). These results were based on 1k simulations of independently drawn data. The λ column corresponds to the parameter used in the HAL fit, which controls the amount of undersmoothing (where λ_{CV} corresponds to no undersmoothing and the smaller values of λ correspond to increased amounts of undersmoothing).

Table 3.3 shows the identical results as Table 3.2 but for sample size of 10k.

	N	λ	true ψ_0	Mean Est	Bias	Var	MSE	SD	t0
1	10000	λ_{CV}	0.2160	0.2176	0.0015692	0.0000323	0.0000348	0.00569	4
2	10000	0.9*λ_{CV}	0.2160	0.2176	0.0015807	0.0000319	0.0000344	0.00565	4
3	10000	0.8*λ_{CV}	0.2160	0.2176	0.0015891	0.0000315	0.0000340	0.00561	4
4	10000	0.7*λ_{CV}	0.2160	0.2176	0.0015948	0.0000310	0.0000336	0.00557	4
5	10000	0.6*λ_{CV}	0.2160	0.2176	0.0015980	0.0000305	0.0000330	0.00552	4
6	10000	0.5*λ_{CV}	0.2160	0.2176	0.0016030	0.0000299	0.0000324	0.00546	4
7	10000	0.4*λ_{CV}	0.2160	0.2176	0.0016057	0.0000292	0.0000318	0.00541	4
8	10000	0.3*λ_{CV}	0.2160	0.2176	0.0015958	0.0000286	0.0000311	0.00535	4
9	10000	0.2*λ_{CV}	0.2160	0.2176	0.0015759	0.0000279	0.0000304	0.00529	4
10	10000	0.1*λ_{CV}	0.2160	0.2176	0.0015224	0.0000272	0.0000295	0.00522	4

Table 3.3: Simulation results for IPCW-HAL estimators across different levels of undersmoothing applies to data sets of size 10k (Independent Resampling). These results were based on 1k simulations of independently drawn data. The λ column corresponds to the parameter used in the HAL fit, which controls the amount of undersmoothing (where λ_{CV} corresponds to no undersmoothing and the smaller values of λ correspond to increased amounts of undersmoothing).

Figure 3.1 shows that the variance of all IPCW+HAL estimators is smaller than the corresponding IPCW estimators, an improvement that becomes more pronounced with increased amounts of undersmoothing (although this won't be true in all cases and more attention should be paid to selecting an optimal L_1 norm in practice).

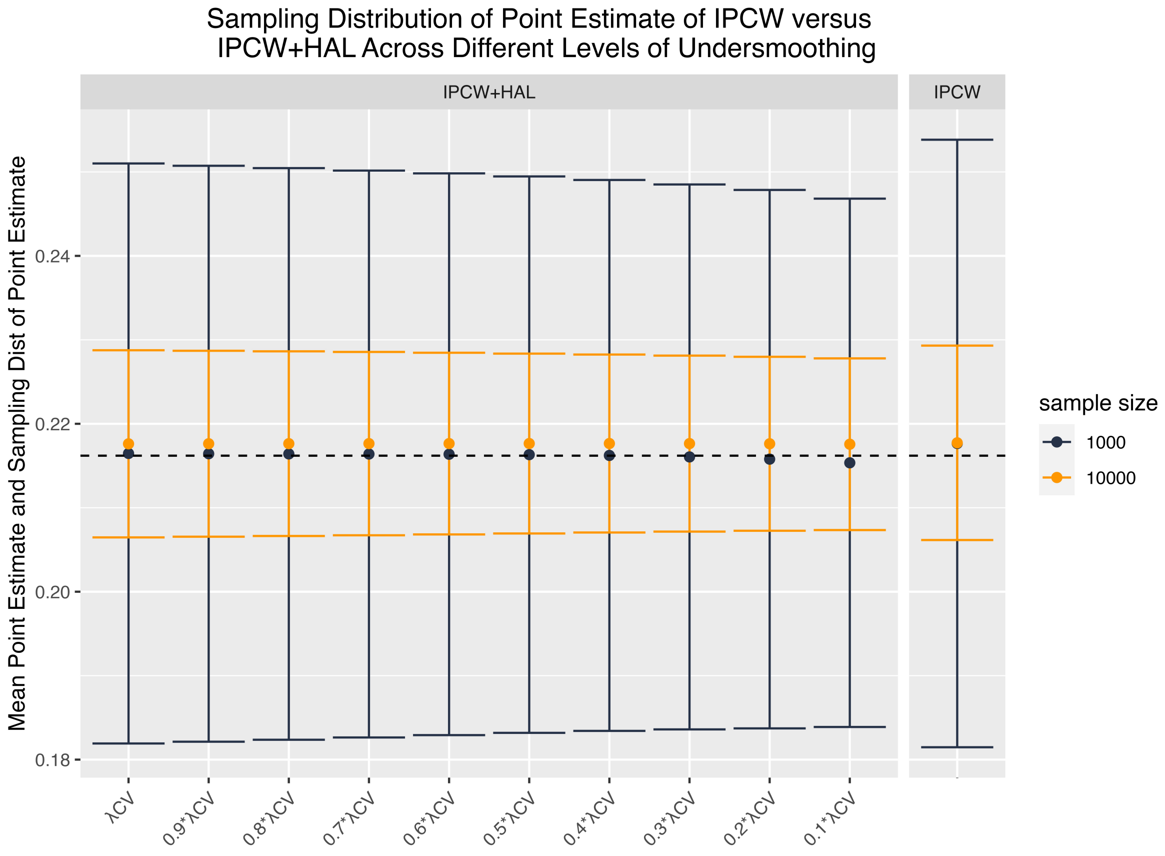

Figure 3.1: Interval plots for the sampling distribution of point estimates from simulations on data of sample sizes 1k and 10k to assess performance of the IPCW estimator using known probabilities for weights versus IPCW+HAL estimators with different degrees of undersmoothing (Independent Resampling).

Figure 3.2 shows the bias of all IPCW and IPCW-HAL estimators. When N=1k, all IPCW+HAL estimators reduce bias relative to the IPCW estimator – again, with additional undersmoothing further reducing this bias. In the 10k sample size, the difference in bias is less pronounced across estimators.

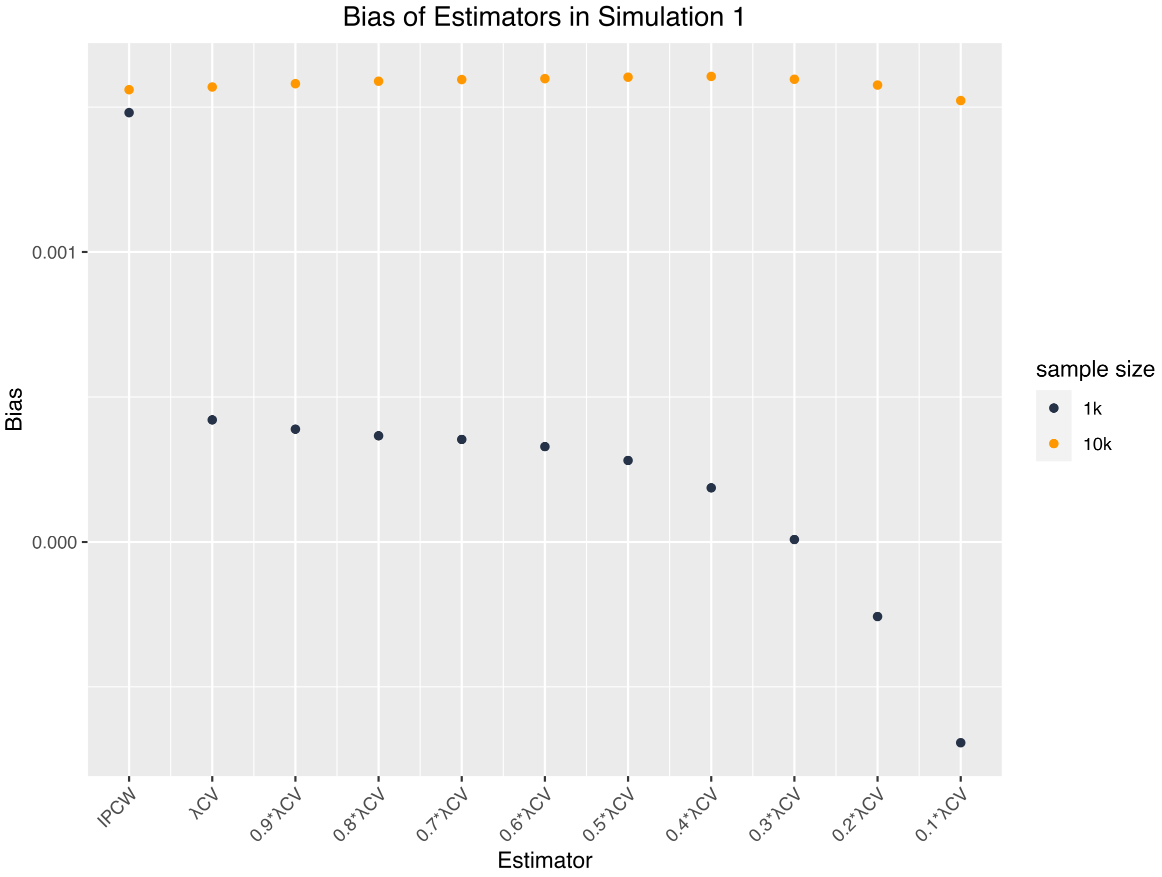

Figure 3.2: Bias of Estimators in Simulation 2 (Independent Resampling). From left to right: IPCW without HAL, IPCW+HAL with no undersmoothing, IPCW+HAL with increasing amounts of undersmoothing.

Simulation 2: Dependent Resampling

We now present the identical tables and graphs as in Simulation 1, but this time, we look at the performance of our estimators when stratified random (instead of completely random) sampling is used. Simulation study 2 shows a 35% decrease in variance between the IPCW and "best" IPCW-HAL estimator on data if size N=1k. Similarly, we see a 23% decrease in variance for sample size N=10k.

Table 3.4 displays the results for the IPCW estimator which uses known probabilities for the weights across all sample sizes. Again, the variance decreases as sample size increases.

	N	true ψ_0	Mean Est	Bias	Var	MSE	SD	t0
1	100	0.2163	0.2175	0.0017116	0.0032365	0.0032395	0.0568905	4
2	1000	0.2163	0.2175	0.0011838	0.0003288	0.0003302	0.0181336	4
3	10000	0.2163	0.2177	0.0014230	0.0000331	0.0000351	0.0057490	4

Table 3.4: Simulation results for the IPCW estimator with known probabilities for weights (Dependent Resampling). These results were based on 10k simulations of independently drawn data.

Table 3.5 contains results for the IPCW-HAL estimators for sample size N=1k. The estimator corresponding to row one uses the cross validated choice of λ and is not undersmoothed at all. As the row index increases, the amount of undersmoothing increases as well. We refer to the estimator in row 10 as the "best" as it has the smallest variance (though in reality, this choice is more nuanced and shoulder consider the change in bias).

	N	λ	true ψ_0	Mean Est	Bias	Var	MSE	SD	t0
1	1000	λ_{CV}	0.2163	0.2254	0.0091176	0.0002748	0.0003579	0.01658	4
2	1000	0.9*λ_{CV}	0.2163	0.2250	0.0087526	0.0002711	0.0003477	0.01647	4
3	1000	0.8*λ_{CV}	0.2163	0.2246	0.0083273	0.0002673	0.0003367	0.01635	4
4	1000	0.7*λ_{CV}	0.2163	0.2241	0.0078277	0.0002636	0.0003249	0.01624	4
5	1000	0.6*λ_{CV}	0.2163	0.2235	0.0072200	0.0002593	0.0003114	0.01610	4
6	1000	0.5*λ_{CV}	0.2163	0.2228	0.0064812	0.0002549	0.0002969	0.01597	4
7	1000	0.4*λ_{CV}	0.2163	0.2219	0.0055892	0.0002509	0.0002822	0.01584	4
8	1000	0.3*λ_{CV}	0.2163	0.2208	0.0045495	0.0002471	0.0002678	0.01572	4
9	1000	0.2*λ_{CV}	0.2163	0.2196	0.0033425	0.0002442	0.0002554	0.01563	4
10	1000	0.1*λ_{CV}	0.2163	0.2183	0.0019999	0.0002439	0.0002479	0.01562	4

Table 3.5: Simulation results for IPCW-HAL estimators across different levels of undersmoothing applies to data sets of size 1k (Dependent Resampling). These results were based on 1k simulations of independently drawn data. The λ column corresponds to the parameter used in the HAL fit, which controls the amount of undersmoothing (where λ_{CV} corresponds to no undersmoothing and the smaller values of λ correspond to increased amounts of undersmoothing).

Table 3.6 shows the identical results as Table 3.5 but for sample size 10k.

	N	λ	true ψ_0	Mean Est	Bias	Var	MSE	SD	t0
1	10000	λ_{CV}	0.2163	0.2194	0.0031524	0.0000297	0.0000397	0.00545	4
2	10000	0.9*λ_{CV}	0.2163	0.2193	0.0030409	0.0000293	0.0000386	0.00542	4
3	10000	0.8*λ_{CV}	0.2163	0.2191	0.0028713	0.0000290	0.0000372	0.00538	4
4	10000	0.7*λ_{CV}	0.2163	0.2190	0.0027087	0.0000286	0.0000359	0.00535	4
5	10000	0.6*λ_{CV}	0.2163	0.2188	0.0025417	0.0000283	0.0000347	0.00532	4
6	10000	0.5*λ_{CV}	0.2163	0.2186	0.0023702	0.0000280	0.0000336	0.00529	4
7	10000	0.4*λ_{CV}	0.2163	0.2185	0.0021910	0.0000278	0.0000326	0.00527	4
8	10000	0.3*λ_{CV}	0.2163	0.2183	0.0020030	0.0000275	0.0000315	0.00524	4
9	10000	0.2*λ_{CV}	0.2163	0.2181	0.0018067	0.0000272	0.0000305	0.00522	4
10	10000	0.1*λ_{CV}	0.2163	0.2179	0.0016203	0.0000270	0.0000296	0.00519	4

Table 3.6: Simulation results for IPCW-HAL estimators across different levels of undersmoothing applies to data sets of size 10k (Dependent Resampling). These results were based on 1k simulations of independently drawn data. The λ column corresponds to the parameter used in the HAL fit, which controls the amount of undersmoothing (where λ_{CV} corresponds to no undersmoothing and the smaller values of λ correspond to increased amounts of undersmoothing).

Figure 3.3 shows that the bias in some IPCW+HAL estimators is larger than what we saw for the independent resampling case. The undersmoothing plays a larger role in bias reduction than in Simulation 1. The variance of all IPCW+HAL estimators is smaller than the IPCW estimators, an improvement that once again becomes more pronounced with increased amounts of undersmoothing.

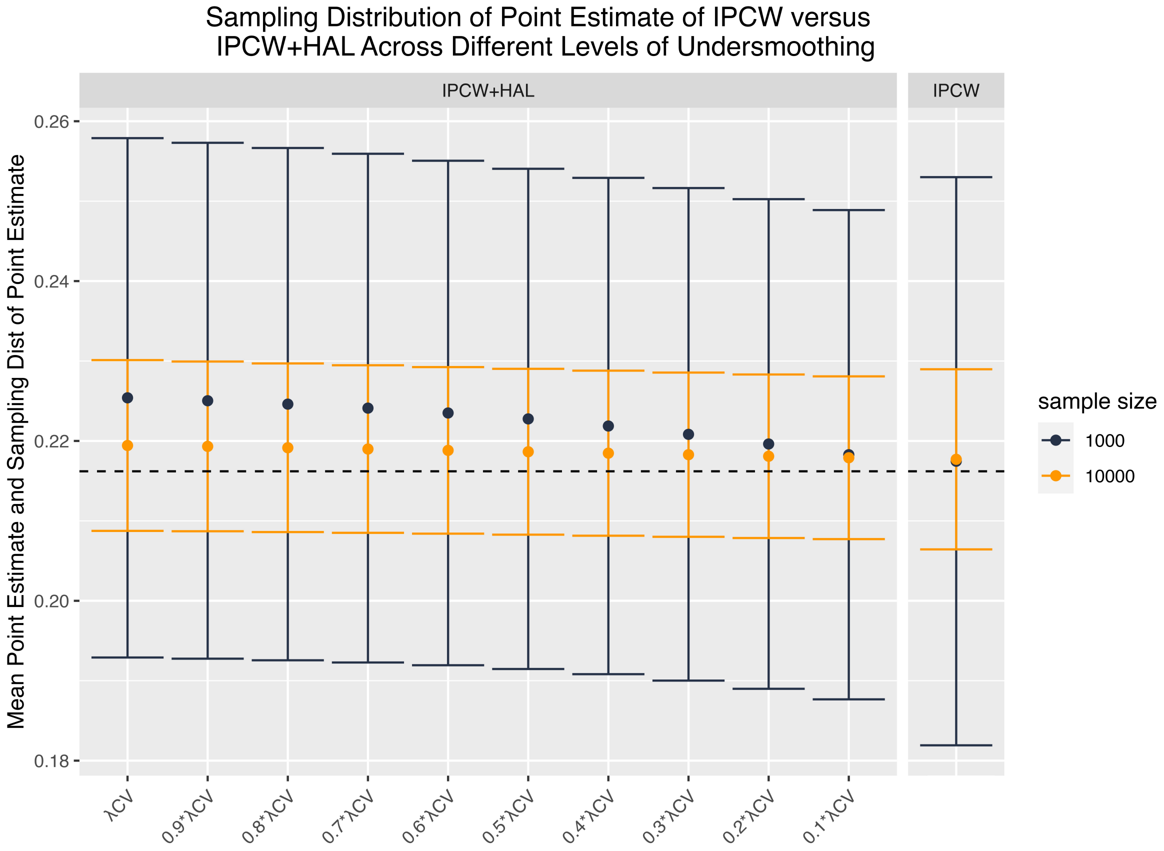

Figure 3.3: Interval plots for the sampling distribution of point estimates from simulations on data of sample sizes 1k and 10k to assess performance of IPCW estimator using known probabilities for weights versus IPCW+HAL estimators with different degrees of undersmoothing (Dependent Resampling).

Figure 3.4 highlights the slightly larger bias of IPCW-HAL estimators in Simulation 2 relative to Simulation 1, specifically among the smaller sample size. This bias is not cause for concern as it remains less than $1/max(log(n), 10)$. This is a rough metric which helps to ensure both good coverage and control over MSE.

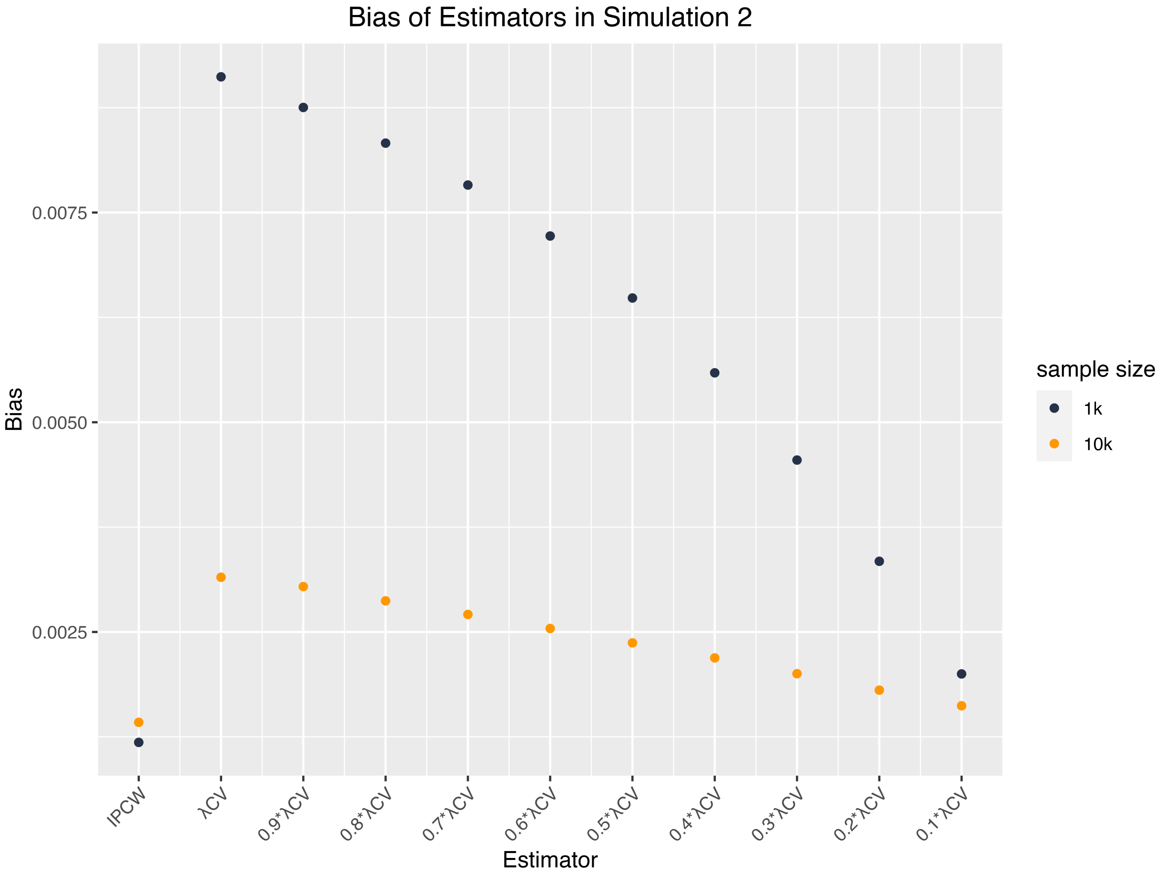

Figure 3.4: Bias of Estimators in Simulation 2 (Dependent Resampling). From left to right: IPCW without HAL, IPCW+HAL with no undersmoothing, IPCW+HAL with increasing amounts of undersmoothing.

Chapter 4

Discussion

The IPCW estimator derived in this work deviates slightly from the typical form of these estimators. One of the interventions we impose sets $\tau \geq t_0$ and prevents right censoring by study end until the time of interest for mortality estimation. The IPCW function contains a sum (or in the continuous case, an integral) over all possible values of τ which satisfy $\tau \geq t_0$. To our knowledge, this deviation has not been discussed in the context of resampling designs (perhaps due to differences in how the "full data" is defined across different works). However, this important deviation should be noted if one wishes to implement IPW estimators in resampling designs and define the full data as having no right censoring by study end.

Our simulations demonstrate that IPCW estimators which make use of undersmoothed HAL estimators for their propensity scores can have approximately 30% smaller variance than IPCW estimators which use known probabilities for their weights; although, with very large sample sizes this difference may be less pronounced. Since IPCW estimators require correct specification of the model for the propensity scores, using a nonparametric estimator like HAL is a safer option than relying on a possibly misspecified parametric model should the true weights not be known by design. IPCW estimators may have been avoided in the analysis of resampling designs due to their potentially high variance, but our application of IPCW-HAL estimators proves that these estimators should not be overlooked but rather augmented with HAL to decrease their variance.

While we have worked to create a flexible framework which allows for the addition of real world complexities, our current implementation of these methods is still limited. One important covariate, a CD4 count tracked over time, was binarized in order to avoid estimation of conditional densities in our plug-in estimator (simulation results left to future work) and to decrease computational burdens. The development and implementation of a HAL estimator for conditional densities in larger data sets is ongoing, and this estimator will be incorporated when software becomes available – at which point we could reintroduce the continuous version of our CD4 count variable [18].

Our approach warrants further development to encompass a broader range of scenarios. Future endeavors involve the identification of an optimal resampling scheme that capitalizes on existing covariate information to further improve efficiency, as exemplified in other studies on resampling designs [11]. We wish to provide extensions to cover cases where resampling occurs more than once or where resampling is incomplete and not all chosen patients are successfully located. Lastly, augmenting any existing estimation approach with HAL is computationally burdensome as HAL "blows up" the data from $n * p$ to $n * n(2^p - 1)$ at a maximum before running lasso regression. This burden can be especially limiting in data with a larger number of covariates, especially if they are continuous.

While the simulations demonstrate the benefit of undersmoothing, we have yet to implement criterion for selecting an optimal amount of undersmoothing. Literature exists on many potential approaches to be explored [14] [5] [6].

This work demonstrates the potential for undersmoothed HAL to improve the efficiency and bias of some of the most commonly used estimators in causal inference while avoiding parametric assumptions and providing an unparalleled rate of convergence among machine learning methods even in high dimensional data. While our estimators may be somewhat limited by computational time, we believe them to be a worthwhile investment given the demonstrated efficiency gains and robustness to complex underlying data structures.

Bibliography

[1] Charles B Holmes, Izukanji Sikazwe, Kombatende Sikombe, Ingrid Eshun-Wilson, Nancy Czaicki, Laura K Beres, Njekwa Mukamba, Sandra Simbeza, Carolyn Bolton Moore, Cardinal Hantuba, et al. Estimated mortality on hiv treatment among active patients and patients lost to follow-up in 4 provinces of zambia: findings from a multistage sampling-based survey. *PLoS medicine*, 15(1):e1002489, 2018.

[2] Elvin H Geng, Thomas A Odeny, Rita E Lyamuya, Alice Nakiwogga-Muwanga, Lameck Diero, Mwebesa Bwana, Winnie Muyindike, Paula Braitstein, Geoffrey R Somi, Andrew Kambugu, et al. Estimation of mortality among hiv-infected people on antiretroviral treatment in east africa: a sampling based approach in an observational, multisite, cohort study. *The lancet HIV*, 2(3):e107–e116, 2015.

[3] Constantin T Yiannoutsos, Ming-Wen An, Constantine E Frangakis, Beverly S Musick, Paula Braitstein, Kara Wools-Kaloustian, Daniel Ochieng, Jeffrey N Martin, Melanie C Bacon, Vincent Ochieng, et al. Sampling-based approaches to improve estimation of mortality among patient dropouts: experience from a large pepfar-funded program in western kenya. *PloS one*, 3(12):e3843, 2008.

[4] James M Robins and Dianne M Finkelstein. Correcting for noncompliance and dependent censoring in an aids clinical trial with inverse probability of censoring weighted (ipcw) log-rank tests. *Biometrics*, 56(3):779–788, 2000.

[5] Ashkan Ertefaie, Nima S Hejazi, and Mark J van der Laan. Nonparametric inverse-probability-weighted estimators based on the highly adaptive lasso. *Biometrics*, 2022.

[6] Mark J van der Laan, David Benkeser, and Weixin Cai. Efficient estimation of pathwise differentiable target parameters with the undersmoothed highly adaptive lasso. *The International Journal of Biostatistics*, 2022.

[7] Maya L Petersen and Mark J van der Laan. Causal models and learning from data: integrating causal modeling and statistical estimation. *Epidemiology (Cambridge, Mass.)*, 25(3):418, 2014.

[8] Gang Li and Chi-hong Tseng. Non-parametric estimation of a survival function with two-stage design studies. *Scandinavian journal of statistics*, 35(2):193–211, 2008.

[9] Constantine E Frangakis and Donald B Rubin. Addressing an idiosyncrasy in estimating survival curves using double sampling in the presence of self-selected right censoring. *Biometrics*, 57(2):333–342, 2001.

[10] James Robins, Andrea Rotnitzky, and Marco Bonetti. Discussion of the frangakis and rubin article. *Biometrics*, 57(2):343–347, 2001.

[11] Ming-Wen An, Constantine E Frangakis, Constantin T Yiannoutsos, et al. Choosing profile double-sampling designs for survival estimation with application to pepfar evaluation. 2015.

[12] Sherri Rose and Mark J van der Laan. A targeted maximum likelihood estimator for two-stage designs. *The international journal of biostatistics*, 7(1), 2011.

[13] Mark J Van der Laan, Sherri Rose, et al. *Targeted learning: causal inference for observational and experimental data*, volume 10. Springer, 2011.

[14] Haodong Li, Sonali Rosete, Jeremy Coyle, Rachael V Phillips, Nima S Hejazi, Ivana Malenica, Benjamin F Arnold, Jade Benjamin-Chung, Andrew Mertens, John M Colford Jr, et al. Evaluating the robustness of targeted maximum likelihood estimators via realistic simulations in nutrition intervention trials. *Statistics in Medicine*, 41(12): 2132–2165, 2022.

[15] Alejandro Schuler and Mark J Van der Laan. *Introduction to Modern Causal Inference*. 2021.

[16] Mark J Laan and James M Robins. *Unified methods for censored longitudinal data and causality*. Springer, 2003.

[17] David Benkeser and Mark Van Der Laan. The highly adaptive lasso estimator. In *2016 IEEE international conference on data science and advanced analytics (DSAA)*, pages 689–696. IEEE, 2016.

[18] Helene Charlotte Wiese Rytgaard, Frank Eriksson, and Mark van der Laan. Estimation of time-specific intervention effects on continuously distributed time-to-event outcomes by targeted maximum likelihood estimation. *arXiv preprint arXiv:2106.11009*, 2021.

Chapter 5

Appendix

Data generation process:

$$
\begin{gathered}
L0 \sim I([N(\mu = 290, \sigma = 60), 10 < L0 < \infty] < 200) \\
\tau \sim M(\pi), \pi = (0.1, 0.1, 0.1, 0.1, 0.1, 0.6) \\
A(0) = I(\tau \leq 0) \\
D_{unobs}(1) = Bern(expit(-1.5L0 - 2.5(1 - L0))) \\
Id(1) = Bern(.9)I(D_{unobs}(1) = 1) \\
D_{obs}(1) = I(D_{unobs}(1) = 1)I(Id(1) = 1) \\
V(1) = Bern(expit(-2.5))I(D_{unobs}(1) = 0)I(A(0) = 0) \\
Lt(1) = I(V(1) = 1)Bern(expit(0.75L0 - 3(1 - L0))) \\
A(1) = I(\tau \leq 1) \\
D_{unobs}(2) = Bern(expit(-.8L0 - 1.6(1 - L0)) - Lt(1) - 1.1(1 - Lt(1)))) \\
Id(2) = Bern(.3)I(D_{unobs}(2) = 1)I(D_{unobs}(t-) = 0) \\
D_{obs}(2) = I(D_{unobs}(2) = 1)I(Id(2) = 1) \\
V(2) = Bern(expit(2.5V(1) - 2(1 - V(1))))I(D_{unobs}(2-) = 0)I(A(1) = 0) \\
Lt(2) = I(V(2) = 1)Bern(expit(0.6L0 - .5(1 - L0) + 2Lt(1) - 2(1 - Lt(1))) \\
A(2) = I(\tau \leq 2) \\
D_{unobs}(3) = Bern(expit(0.05L0 - .5(1 - L0) + .3Lt(1) - .7(1 - Lt(1)) - 1.8Lt(2) - \\
2.5(1 - Lt(2)))) \\
Id(3) = Bern(.3)I(D_{unobs}(3) = 1)I(D_{unobs}(t-) = 0) \\
D_{obs}(3) = I(D_{unobs}(3) = 1)I(Id(3) = 1) \\
V(3) = Bern(expit(0.2V(1) - 0.2(1 - V(1)) + 2.3V(2) + 0.6(1 - V(2)))) \\
I(D_{unobs}(3-) = 0)I(A(2-) = 0) \\
Lt(3) = I(V(3) = 1)Bern(expit(0.5L0 - 0.5(1 - L0) + 0.7Lt(1) - 0.5(1 - Lt(1)) + \\
2Lt(2) + (1 - Lt(2)))
\end{gathered}
$$

$$D_{unobs}(4) = Bern(expit(0.05L0 - 0.2(1 - L0) + 0.05Lt(1) - 0.2(1 - Lt(1)) - 0.1Lt(2) -$$

$$0.7(1 - Lt(2)) - Lt(3) - 2.5(1 = Lt(3))))$$

$$Id(4) = Bern(.3)I(D_{unobs}(4) = 1)I(D_{unobs}(t-) = 0)$$

$$D_{obs}(4) = I(D_{unobs}(4) = 1)I(Id(4) = 1)$$

$$V(4) = Bern(expit(0.2V(1) - 0.2(1 - V(2)) + 2.3V(3) + 0.6(1 - V(3))))$$

$$I(D_{unobs}(4-) = 0)I(A(3-) = 0)$$

$$Lt(4) = I(V(4) = 1)Bern(expit(0.5L0 - 0.7(1 - L0) + 0.6Lt(1) - 0.5(1 - Lt(1)) +$$

$$0.8Lt(2) + 0.5(1 - Lt(2)) + 2.5Lt(3) - (1 - Lt(3)))$$

$$D_{unobs}(5) = Bern(expit(0.02L0 - 0.2(1 - L0) + 0.05Lt(1) - 0.1(1 - Lt(1)) - 0.5Lt(2) -$$

$$0.1(1 - Lt(2)) + 0.1Lt(3) - 0.3(1 = Lt(3)) - Lt(4) - 2(1 - Lt(4))))$$

$$Id(5) = Bern(.3)I(D_{unobs}(5) = 1)I(D_{unobs}(t-) = 0)$$

$$D_{obs}(5) = I(D_{unobs}(5) = 1)I(Id(5) = 1)$$

$$V(5) = Bern(expit(0.1V(1) + 0.2V(2) - 0.05(1 - V(2)) + 0.2V(3) - 0.2(1 - V(3)) +$$

$$2.3V(4) + 0.6(1 - V(4))))$$

$$Lt(5) = I(V(5) = 1)Bern(expit(0.1Lt(1) - 0.5(1 - Lt(1)) + 0.1Lt(2) - 0.5(1 - Lt(2)) +$$

$$0.2Lt(3) - 0.5(1 - Lt(3)) + 2.5Lt(4) - 2.5(1 - Lt(4)))$$