Erasmus University Rotterdam

Master Thesis

MSc. Econometrics and Management Science

Business Analytics and Quantitative Marketing

An Iterative Hard Thresholding Method for Feature Selection in Patient-Level Prediction Models

Mathilde Tans

446685

Erasmus School of Economics

Supervisor: Dr. E.P. O'Neill

Second assessor: Dr. W. Wang

Erasmus MC
Universitair Medisch Centrum Rotterdam

Supervisor: R. Williams

Supervisor: Dr. P. Rijnbeek

April 24, 2022

Abstract

With a need for automatic selection of parsimonious models in clinical prediction modeling, we develop an iterative hard thresholding (IHT) method to integrate with the Patient-Level Prediction (PLP) methodology. The method maximises the ℓ_0-penalised log-likelihood and selects a small number of important predictors, while simultaneously estimating the corresponding coefficient values. In order to unite the PLP methodology with the IHT algorithm, we deviate from the standard IHT procedure by employing cyclic coordinate descent (CCD) and incorporate a ridge penalty for each cycle in order to achieve convergence. Additionally, we make use of "warm starts" and introduce two extensions, step-halving and screening, with the primary purpose of decreasing the estimation time. We derive the algorithm for Logistic Regressions and Cox Proportional Hazard Models. The variable selection property of the algorithm is verified in simulation, after which the novel algorithm is compared to the current benchmark for sparse estimation in clinical prediction modeling, the lasso algorithm, in a real real data application. The IHT algorithm is evaluated on its predictive performance by assessing the discrimination and calibration. We find that IHT outperforms lasso, i.e., IHT remains close to the optimal lasso and achieves significantly higher performance when lasso selects a similar number of covariates, and lastly, that the novel algorithm retains its predictive performance in external validation.

List of Abbreviations

AFF atrial fibrillation or flutter

AUC area under the receiver operating characteristic

BIC Bayesian information criterion

CCD cyclic coordinate descent

CDM common data model

Cox Cox Proportional Hazard

EHR electronic health record

GLM generalized linear model

IHT iterative hard thresholding

IPCI Integrated Primary Care Information

LR Logistic Regression

OHDSI Observational Health Data Sciences and Informatics

OMOP Observational Medical Outcomes Partnership

PLP Patient-Level Prediction

Contents

1 Introduction

The revolution that followed from the introduction of the Internet, and a bit later, the smartphone, is arguably comparable to the revolution caused by the printing press (Topol, 2015). The revolutions share three important characteristics: (1) the explosion of knowledge; (2) the opportunity for the individual to access it; and (2) for that knowledge to be spread at unprecedented speeds. Recent years, and especially the current COVID-19 era, have shown the vast implications this revolution can have for the health care industry, an important one being the global network of digital observational evidence that has materialised (Murdoch & Detsky, 2013; OHDSI, 2021). This network is structured by the Observational Medical Outcomes Partnership (OMOP) common data model (CDM), and it provides a mapping between disparate databases. Such databases consist of electronic health records (EHRs), which contain an individual's patient profile.

This global network can be leveraged in clinical prediction problems, and the CDM allows us to perform transparent and verifiable research. This enables the safe implementation of prediction problems in clinical practice (OHDSI, 2021). Clinical predictions support the clinical decision-making process, and are based on a combination of patient characteristics, of which there are many to choose from (OHDSI, 2021). The number of covariates that are derived from these patient characteristics can grow into the tens of thousands to hundreds of thousands, which is too many for clinicians to keep track of, and hence, forms an obstacle to clinical implementation. Currently, we need a physician's expertise to select the covariates. While it produces reasonable models, this approach is not scalable. Therefore, there is need for a method to automatically select parsimonious models with ten to twenty covariates.

This leaves us with a variable selection problem. The current default method for variable selection in clinical prediction modeling is the lasso algorithm (Steyerberg, 2019). However, to obtain a model with good predictive performance, lasso selects many covariates into the active set. Among these covariates are the true predictors, but lasso also includes noise and correlated covariates (Bertsimas, King, & Mazumder, 2016). If one would increase the penalty, less covariates would be nonzero, but at the cost of leaving out true predictors. Since clinical prediction data is

correlated in general (Steyerberg, 2019), the lasso algorithm is not suited to select only a handful of covariates from tens to hundreds of thousands if you want to achieve good predictive accuracy.

Recently, promising results were found for a method called iterative hard thresholding (IHT), which uses a combination of gradient descent and a hard thresholding operator to find a sparse estimation in an iterative fashion (Chu et al., 2020). This method optimises the ℓ_0-regularised log-likelihood, i.e., automatically selects a subset of covariates and simultaneously estimates their coefficients (Blumensath & Davies, 2009). Since the hard threshold allows the user to set an upperbound on the number of selected covariates and the algorithm is scalable to high-dimensional data sets (Xu & Chen, 2014), this method is an attractive potential candidate for a feature selection algorithm in clinical predictive modeling.

This research builds upon the foundation laid by the Observational Health Data Sciences and Informatics (OHDSI) network[1] with their work on Patient-Level Predictions (PLPs)—the OHDSI term for clinical prediction models, and we primarily focus on developing an iterative hard thresholding feature selection algorithm for PLP models. Our contribution is threefold. Firstly, we adapt the current IHT algorithm such that it can handle the high-dimensionality and sparsity found in PLP data. Secondly, we introduce two extensions, whose goals are to decrease the estimation time. The third contribution concerns the external validation of the various models, since it is the first time that an automated feature selection algorithm for PLP models is externally validated.

The different versions of the novel algorithm, i.e., the IHT algorithm with and without extensions, are derived for Logistic Regressions (LRs) and Cox Proportional Hazard (Cox) Regressions. The algorithm and its variable selection property are evaluated during a simulation study, where we confirm that the IHT algorithm has the ability to select the nonzero covariates without including any zero covariates. Subsequently, the algorithm is applied to a real data set, where where the risk of all-cause mortality is predicted for patients with atrial fibrillation or flutter (AFF). The novel algorithms are compared to lasso and the predictive performance is assessed in terms of discrimination and calibration. Firstly, IHT's performance closely approaches that of lasso when lasso is optimised, in which case the number of nonzero covariates is significantly lower for IHT.

[1] https://www.ohdsi.org/.

Secondly, when lasso includes a similar number of nonzero covariates, the IHT algorithms obtain substantially better predictive performance. Additionally, we find that these results carry over in external validation.

The paper is structured as follows. In Section 2, we review related work. In Section 3, we elaborate on the PLP framework, which more clearly explains the context of this research. Then, in Section 4, we outline the methodology and introduce the feature selection algorithm developed for PLP models. The methods are tested in simulations studies in Section 5, and applied to a real data set in Section 6. Lastly, the conclusion is presented in Section 7.

2 Related Work

The IHT algorithm was formally introduced by Blumensath and Davies (2009) for linear regressions, and the algorithm finds its origin in signal approximations (Herrity, Gilbert, & Tropp, 2006). Since then, the authors of the original algorithm developed IHT further to provide convergence guarantees and to accelerate the estimation. This resulted in normalised IHT (Blumensath & Davies, 2010), and accelerated IHT (Blumensath, 2012), respectively. The method gained traction in feature screening research, where the least important covariates are screened to then be removed from the model. Xu and Chen (2014) implemented IHT for feature screening and derived the algorithm for generalized linear models (GLMs). Yang, Yu, Li, and Buu (2016) built upon the work of Xu and Chen (2014) and extended the IHT algorithm to Cox models, whereas Z. Liu and Xiong (2022) did the same for additive hazard models. Then, X. Chen, Liu, and Xu (2021) used lasso as an initial estimate to their IHT method for Cox models. They employed an IHT algorithm similar to those of Xu and Chen (2014) and Yang et al. (2016), but improved by selecting the step size in gradient descent in a more advanced fashion. The work in Y. Liu, Xu, and Li (2021) is comparable to X. Chen et al., 2021, in the respect that they also developed a lasso-initiated IHT algorithm, although they derived the method for right-censored data, where Cox models are a special case.

In work similar to our IHT algorithm, Zheng, Fan, and Lv (2014) investigated some hard threshold and an ℓ_0-regularisation for linear regressions, where they also explored the possibility of

including further ℓ_2-regularisations. Z. Liu, Sun, and McGovern (2017) provided an algorithm for ℓ_0 approximations that includes the iterative, element-wise ridge-regularisation for GLMs in high-dimensional biomedical data. By setting low penalties for important predictors and high penalties for unimportant predictors, their estimation should approximate the ℓ_0-penalised regression when the number of iterations goes to infinity. Kawaguchi, Suchard, Liu, and Li (2020) introduce a similar method as Z. Liu et al. (2017) for high-dimensional time-to-event data. Secondly, Su, Wijayasinghe, Fan, and Zhang (2016) and Y. Chen and Zhao (2021) approach the ℓ_0-norm penalty and feature selection from the approximation of an information criterion, e.g., the Bayesian information criterion (BIC), where Su et al. (2016) derive their algorithm for Cox models, and Y. Chen and Zhao (2021) for interval-censored data.

For the interested reader, other ℓ_0-norm oriented feature selection algorithms are based on augmented and penalised minimisation for censored data in Li, Xie, Zeng, and Wang (2018), and on second-generation P-values in Zuo, Stewart, and Blume (2021). Furthermore, in the field of subset selection or sparse estimation, promising and related research has been presented in Bertsimas et al. (2016) who implemented mixed integer optimisation for best subset selection, and Bertsimas and Van Parys (2020) who introduced a novel cutting plane technique. Lastly, interesting work has been performed in Erion et al. (2021) who utilised artificial intelligence to select a subset of covariates, or Hazimeh and Mazumder (2020) who used a method that shares similarities with IHT but also incorporates a combinatorial approach.

3 Patient-Level Prediction

The objective of a patient-level prediction model is to find an answer to the general question: 'Among a specific group of patients, who is at risk of experiencing some clinical outcome during the time-at-risk period?' The specific group of patients is called the target cohort, and the time these patients enter this cohort defines the start of the time-at-risk. The target cohort can be, for instance, the group of patients that are (1) newly diagnosed with some disease, (2) recently started some medication, or (3) just went through a certain procedure (Reps, Schuemie, Suchard,

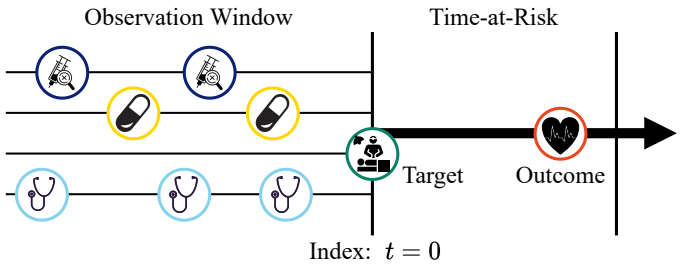


Figure 1. The Patient-Level Prediction (PLP) problem. The target, which is a procedure here, defines the index date. The index date is the start of the time-at-risk. The objective is to find the probability of the outcome occurring during the time-at-risk.

Ryan, & Rijnbeek, 2018). Then, from the moment a patient enters the target cohort, also known as the 'index date', the time-at-risk starts. The time-at-risk can be one month, but it can also be a few years. Finally, the group of patients that has the outcome is denoted as the outcome cohort. Figure 1 shows an illustration of the prediction problem and how, for one patient, the target—which in this case is some procedure, determines the start of the time-at-risk. The objective is to find the probability that the outcome will occur during the time-at-risk. An illustration of the way the target and outcome cohorts are constructed from patient records, for six patients, is shown in Figure 2, where the target is some procedure and the outcome is represented by the heart icon.

The question posed before is usually answered by developing a risk score using certain patient characteristics. These characteristics could indicate that a patient is at a higher risk of experiencing the clinical outcome, and they can include:

- Demographics characteristics, such as age, gender, race, or index month;

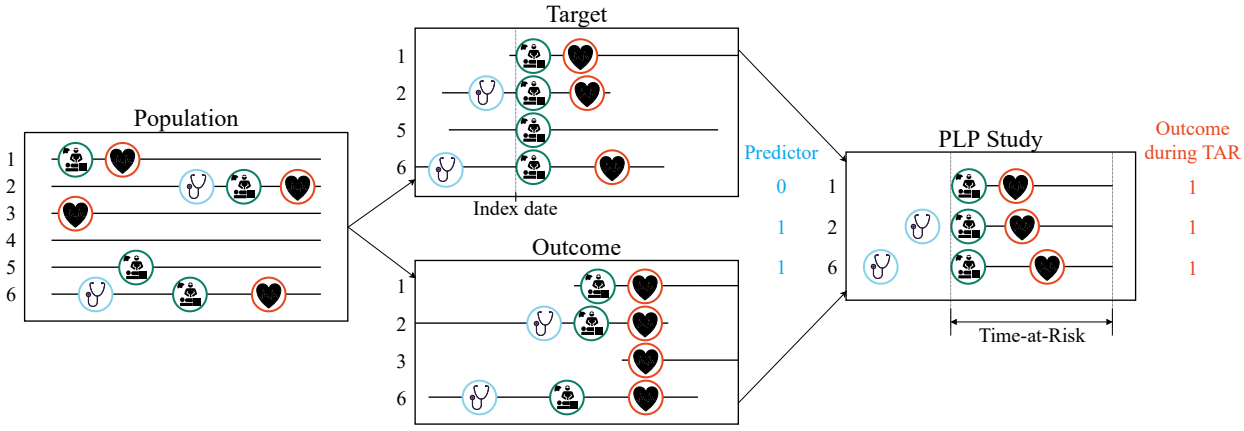


Figure 2. The preparation of a data set for a PLP analysis. The target and outcome cohorts are created separately and then combined for the PLP study.

- All conditions, drugs, measurements, procedures, and observations recorded within the last n days before the index date;
- Hierarchical groupings of the conditions, drugs, measurements, procedures, and observations.

The OHDSI network builds its own software for estimating the PLP models[2]. The software and methodology is developed specifically for PLP data, which are retrieved from longitudinal observational databases that provide time-stamped patient-level medical information. Examples of databases are medical insurance claims databases or databases containing EHRs. PLP data is characterised by high-dimensionality and sparseness. Namely, there are enormous amounts of conditions a patient can be diagnosed with, but a single patient is likely to be diagnosed with only a few (Suchard, Simpson, Zorych, Ryan, & Madigan, 2013).

4 Methods

In this section we provide an overview of the IHT methodology and outline how one can incorporate IHT in PLP models. Subsequently, we propose two extensions for the baseline IHT algorithm. We conclude with descriptions of the various evaluation measures.

4.1 Iterative Hard Thresholding

To illustrate the IHT algorithm, assume we perform a Logistic Regression (LR) for the prediction of some medical outcome. Whether patient i experiences the medical outcome within the time-at-risk is presented as y_i, where y_i is binary and $i = 1, \ldots, n$. A patient has m items of medical information, where x_{ij} carries the information for medical item j, $j = 1, \ldots, m$, for patient i. With $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{im})$, $\mathbf{x}_i$ contains the patient profile for patient i. The m covariates are accompanied by coefficient vector $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_m)'$. Then, the probability that patient i will experience the medical outcome within the time-at-risk is specified as

$$\Pr[Y_i = 1] = \frac{\exp(\mathbf{x}_i\boldsymbol{\beta})}{1 + \exp(\mathbf{x}_i\boldsymbol{\beta})}, \tag{1}$$

[2]See https://github.com/OHDSI, and in particular their PLP package.

and, trivially, $\Pr[Y_i = 0] = 1 - \Pr[Y_i = 1]$. For the IHT algorithm, we will mostly be dealing with the log-likelihood. We show the derivation for LRs, however the algorithm is applicable to GLMs as well. Therefore, we take the GLM notation for the LR log-likelihood,

$$\mathcal{L}(\boldsymbol{\beta}) = \sum_{i=1}^{n} y_i \left(\mathbf{x}_i'\boldsymbol{\beta}\right) - \log\left[1 - \exp\left(\mathbf{x}_i\boldsymbol{\beta}\right)\right] . \tag{2}$$

The primary characteristic of the algorithm is to select a subset of k covariates, which is particularly useful when the number of covariates is very large. The IHT objective is be the following,

$$\min -\mathcal{L}(\boldsymbol{\beta}) \quad \text{subject to} \quad \|\boldsymbol{\beta}\|_0 \leq k\,, \tag{3}$$

where $\|\boldsymbol{\beta}\|_0 = \sum_{j=1}^{m} I(\beta_j \neq 0)$, and $I(\cdot)$ is the indicator function.

In short, IHT solves the objective by performing gradient descent and limiting the number of nonzero coefficients to k at each iteration. The algorithm finds an optimum by iteratively going in the direction of the steepest descent, which is the negative gradient $\nabla\mathcal{L}(\boldsymbol{\beta}^{(q)})$, at iteration q. The estimate is updated by $\boldsymbol{\beta}^{(q+1)} = \boldsymbol{\beta}^{(q)} + s^{(q)}\nabla\mathcal{L}(\boldsymbol{\beta}^{(q)})$, where $s^{(q)} > 0$ is the step size.

In the second step we apply the hard thresholding operator, $H_k(\boldsymbol{\beta})$, where the largest k elements of $\boldsymbol{\beta}$ keep their value and all others are set to zero. The j-th element of the thresholding operator is defined as,

$$H_{k,j}(\boldsymbol{\beta}) = \begin{cases} \beta_j & \text{if } |\beta_j| \geq \lambda_k, \\ 0 & \text{if } |\beta_j| < \lambda_k\,. \end{cases} \tag{4}$$

The parameter λ_k equals the smallest value of the k largest elements of $\boldsymbol{\beta}$. Then, the hard thresholding operator returns $H_k(\boldsymbol{\beta}) = [H_{k,1}(\boldsymbol{\beta}), H_{k,2}(\boldsymbol{\beta}), \ldots, H_{k,m}(\boldsymbol{\beta})]'$. As a result, the IHT algorithm approximates (3) by updating the parameter vector $\boldsymbol{\beta}$ at each iteration with

$$\boldsymbol{\beta}^{(q+1)} = H_k\left(\boldsymbol{\beta}^{(q)} + s^{(q)}\nabla\mathcal{L}\big(\boldsymbol{\beta}^{(q)}\big)\right) . \tag{5}$$

The step size can be user specified. However, convergence is only guaranteed when the *restricted*

isometry property is satisfied (Blumensath & Davies, 2009). Certifying whether this is the case can be quite complicated (Bandeira, Dobriban, Mixon, & Sawin, 2013). Therefore, Blumensath and Davies (2009) introduced an optimal step size which should ensure stable performance. Unfortunately, the calculation for the optimal step size requires inverting the Hessian matrix, which is a computationally expensive exercise for high-dimensional matrices (Suchard et al., 2013).

4.2 Patient-Level Prediction: Cyclic Coordinate Descent

Firstly, it should be noted that the setup of PLP models takes a Bayesian approach. The need for taking this approach concerns the parallelisation of expensive computations, which is outside the scope of this research but is explained in more detail in Suchard et al. (2013). For this research, the only relevant implication is that we define prior distributions for the coefficient instead of penalties for penalised log-likelihoods. Moreover, the analyses performed here are not Bayesian and, hence, we also do not report posterior distributions.

Then, we start by assuming a prior for the coefficient vector $\boldsymbol{\beta}$, denoting the prior distribution as $p(\boldsymbol{\beta})$ (Suchard et al., 2013). When the prior follows the Laplace distribution with location parameter $\mu = 0$ and scale parameter $b = \sigma_\beta^2$, it is straightforward to confirm that finding the maximum a posteriori point-estimates is equal to performing a lasso-regularised regression. Similarly, when the prior assumes the Normal distribution with the same parameters, finding the posterior is equal to performing a ridge-regularised regression. The variance defined for both priors is proportionally inverse to the commonly reported penalty of lasso- or ridge-regularised regressions.

PLP models can grow rather large, with the number of patients in the millions and the number of covariates in the ten thousands. For such models, gradient descent is an expensive and inefficient optimisation method. Suchard et al. have found competitive performance using cyclic coordinate descent (Suchard et al., 2013). This optimisation method updates the coefficient vector element-wise, by optimising $\boldsymbol{\beta}$ in one direction while keeping the other elements constant.

An update of the coefficient vector for element j during iteration q is then as follows,

$$\boldsymbol{\beta}^{(q+1)} = \boldsymbol{\beta}^{(q)} + \left(\Delta\beta_j^{(q)}\right)\mathbf{e}_j\,, \tag{6}$$

where the step size is incorporated in the direction of the update, $\Delta\beta_j$:

$$\Delta\beta_j = -\frac{\frac{\partial}{\partial\beta_j}\left[\mathcal{L}\left(\boldsymbol{\beta}\right) + \log p\left(\boldsymbol{\beta}\right)\right]}{\frac{\partial^2}{\partial\beta_j^2}\left[\mathcal{L}\left(\boldsymbol{\beta}\right) + \log p\left(\boldsymbol{\beta}\right)\right]} = -\frac{g_j\left(\boldsymbol{\beta}\right) + \frac{\partial}{\partial\beta_j}\log p\left(\boldsymbol{\beta}\right)}{h_j\left(\boldsymbol{\beta}\right) + \frac{\partial^2}{\partial\beta_j^2}\log p\left(\boldsymbol{\beta}\right)}. \tag{7}$$

The terms $g_j(\boldsymbol{\beta})$ and $h_j(\boldsymbol{\beta})$ are the uni-dimensional gradient and hessian, respectively.

The expression above is written with the logistic log-likelihood in mind. PLP models can also be specified by taking a survival approach. In that case, Cox models are used. The optimisation method is exactly the same, except that the appropriate likelihood and priors are used to obtain lasso- and ridge-regularised Cox regressions. For the specifications, see Mittal, Madigan, Burd, and Suchard (2014). Note, ties are handled by adding a small random quantity to the event times. For a full description of the methodological framework, we refer the reader to Suchard et al. (2013) for GLM models and Mittal et al. (2014) for Cox models.

To reiterate, the current standard strategy to automatically obtain a subset of covariates is lasso. However, the lasso penalty suppresses large coefficients more severely than smaller coefficients, which means the coefficients of true predictors are underestimated to a larger extent than the coefficients of noise or correlated covariates. As a result, it is less likely that true predictors can be distinguished from noise and correlated variables. Hence, to capture most or all of the true predictors, lasso will need to capture much noise and many correlated covariates. As PLP data generally contains correlated covariates (Steyerberg, 2019), lasso is unlikely to be a suitable method to select only five or ten covariates for PLP models.

4.3 Iterative Hard Thresholding for Patient-Level Prediction

Due to the high-dimensionality of PLP models, we cannot just apply the IHT algorithm as described in Section 4.1. Here, we present the novel methodology to incorporate IHT into the existing PLP framework.

We use the notation from Section 4.1. For the novel IHT algorithm, we use the Laplace prior as well as the Normal prior for $\boldsymbol{\beta}$. To distinguish the use of either prior, let us denote the two priors as $p_L(\boldsymbol{\beta}) \sim \text{Laplace}(0, \Sigma_\beta)$ and $p_N(\boldsymbol{\beta}) \sim \text{Normal}(0, \Sigma_\beta)$, where $\Sigma_\beta = \text{diag}(\sigma^2_{\beta,1}, \sigma^2_{\beta,2}, \ldots, \sigma^2_{\beta,m})$. Then,

we define the cyclic coordinate descent updates as

$$\Delta\beta_{j,L}(\boldsymbol{\beta}) = -\frac{g_j\left(\boldsymbol{\beta}\right) + \frac{\partial}{\partial\beta_j}\log p_L\left(\boldsymbol{\beta}\right)}{h_j\left(\boldsymbol{\beta}\right) + \frac{\partial^2}{\partial\beta_j^2}\log p_L\left(\boldsymbol{\beta}\right)}, \quad \text{and} \quad \Delta\beta_{j,N}(\boldsymbol{\beta}) = -\frac{g_j\left(\boldsymbol{\beta}\right) + \frac{\partial}{\partial\beta_j}\log p_N\left(\boldsymbol{\beta}\right)}{h_j\left(\boldsymbol{\beta}\right) + \frac{\partial^2}{\partial\beta_j^2}\log p_N\left(\boldsymbol{\beta}\right)}. \tag{8}$$

We cannot refrain from using an ℓ_1-norm or ℓ_2-norm penalty. Doing so can be computationally intractable (Suchard et al., 2013). Therefore, we are forced to add a penalty. We choose the ℓ_2-norm penalty, i.e., perform a ridge regression, according to popular use in recent literature (Kawaguchi et al., 2020; Z. Liu et al., 2017; Zheng et al., 2014).

Secondly, we initialise the algorithm with "warm starts", i.e., we initialise with the estimated coefficients from a lasso-regularised regression. Lasso is not quite as suited for variable selection when the task is to select a very small fraction of the covariates, since the algorithm can only select the true predictors by including many correlated predictors (Bertsimas et al., 2016). However, the algorithm will estimate the least important covariates as zero. Therefore, lasso is a great candidate to provide the starting values. This is also evidenced by the lasso initialisations in X. Chen et al. (2021); Y. Liu et al. (2021). The baseline IHT algorithm for PLP models reads as follows,

$$\boldsymbol{\beta}^{(1)} = \big(\Delta\beta_{j,L}(\mathbf{0})\big)\mathbf{e}_j, \qquad \forall j = 1,\ldots,m, \quad \rightarrow \boldsymbol{\beta}^{(1)} = H_k\big(\boldsymbol{\beta}^{(1)}\big), \tag{9}$$

$$\boldsymbol{\beta}^{(q+1)} = \boldsymbol{\beta}^{(q)} + \big(\Delta\beta_{j,N}(\boldsymbol{\beta}^{(q)})\big)\mathbf{e}_j, \quad \forall j = 1,\ldots,m, \quad \rightarrow \boldsymbol{\beta}^{(q+1)} = H_k\big(\boldsymbol{\beta}^{(q+1)}\big), \tag{10}$$

where the iteration counter q starts at 1. The algorithm stops when the coefficient estimates is converged. This is achieved when the absolute difference between the new and current locations is 10^{-8} at most, i.e., $\max\{|\boldsymbol{\beta}^{(q+1)} - \boldsymbol{\beta}^{(q)}|\} < 10^{-8}$. The variance for the Laplacian prior is user specified and shall henceforth be referred to as the initialising variance. The variances for all coefficients are the same, hence $\Sigma_\beta = \sigma_\beta^2 I$. Note that this variance is proportionally inverse to the standard lasso penalty.

The variances for the Normal prior are updated with each iteration. The ridge regression is related to the ℓ_0-regularised optimisation problem through the BIC-score. Note also the link between these concepts in Y. Chen and Zhao (2021); Su et al. (2016). The BIC-score for the ridge

regression at iteration q is equal to BIC $= -2\mathcal{L}(\boldsymbol{\beta}) + \log(n)\|\boldsymbol{\beta}^{(q)}\|_0$, where $\|\boldsymbol{\beta}^{(q)}\|_0$ is the number of parameters that is being estimated as nonzero, as well as the ℓ_0-penalty. Hence, the BIC-score formulation is equal to ℓ_0-regularised regressions if the ℓ_0-penalty is set to $\log(n)/2$. As a result the variances for the Normal priors at each iteration q are calculated as,

$$\sigma^2_{\beta,j}(q) = \frac{\left(\beta_j^{(q)}\right)^2}{\log(n)/2}\,. \tag{11}$$

Last, note that larger coefficient estimates are penalised less heavily than smaller coefficients, similar to the work in Z. Liu et al. (2017).

4.4 Extensions

Here we propose two extensions that could improve the methodology, primarily in terms of efficiency, i.e., to decrease the estimation time.

4.4.1 Step-halving

The authors of the original IHT publication have developed the algorithm further since its introduction. The first major update was normalised IHT (Blumensath & Davies, 2010), in order to guarantee convergence. The second development is called accelerated IHT, primarily meant to achieve faster convergence (Blumensath, 2012). Accelerated IHT is defined as any IHT method that finds a decrease in the (penalised) log-likelihood at each iteration. A possible approach is to perform a line search, as in Bertsimas et al. (2016). This translates to an optimisation of the log-likelihood along the line between the current location and the potential new location of $\boldsymbol{\beta}$. Chu et al. (2020) employ a computationally more attractive approach, namely *step-halving*. The algorithm by Chu et al. is designed for gradient descent, but can easily be implemented for CCD.

We introduce a step size multiplier γ and denote a potential new location of $\boldsymbol{\beta}$ as $\boldsymbol{\beta}^{(q+1)*}$. The potential new location only becomes the new location if there is a descent in the penalised log-likelihood. Otherwise the step size multiplier is set to $\gamma = \frac{1}{2}$, which cuts the step in half, and a new potential location is estimated. If there is no descent, the step size multiplier is halved again.

Algorithm 1: Step-halving for iterative hard thresholding

Data: Current location $\boldsymbol{\beta}^{(q)}$; step size multiplier γ
Result: New location $\boldsymbol{\beta}^{(q+1)}$

Initialise step size multiplier as $\gamma = 1$
for $j = 1, \ldots, m$ **do**
  $\boldsymbol{\beta}^{(q+1)*} \leftarrow \boldsymbol{\beta}^{(q)} + \big(\Delta\beta_{j,N}(\boldsymbol{\beta}^{(q)})\big)\mathbf{e}_j$
while $\mathcal{L}\big(\boldsymbol{\beta}^{(q+1)*}\big) + \log p\big(\boldsymbol{\beta}^{(q+1)*}\big) \geq \mathcal{L}\big(\boldsymbol{\beta}^{(q)}\big) + \log p\big(\boldsymbol{\beta}^{(q)}\big)$ ***or*** $\gamma \geq 2^{-5}$ **do**
  $\gamma \leftarrow \frac{\gamma}{2}$
  for $j = 1, \ldots, m$ **do**
    $\boldsymbol{\beta}^{(q+1)*} \leftarrow \boldsymbol{\beta}^{(q)} + \gamma\big(\Delta\beta_{j,N}(\boldsymbol{\beta}^{(q)})\big)\mathbf{e}_j$
$\boldsymbol{\beta}^{(q+1)} \leftarrow H_k\big(\boldsymbol{\beta}^{(q+1)*}\big)$

The step size multiplier can be halved five times at most. If there is still no descent after these five steps, the latest potential location becomes the new location. The procedure is described in Algorithm 1.

4.4.2 Screening

As was already established, theory suggests that the lasso algorithm is an appropriate method to provide the starting values to the IHT algorithm. Inspired by Fan, Gong, and Sun (2021), we take the lasso initialisation a step further by using it for a screening step. That is, the screening extension starts with a lasso-regularised regression. The variance that is given to the Laplace prior for this regression is referred to as the screening variance, denoted by ς^2. Note that again the prior variance is equal for all elements, hence $\Sigma_\beta = \varsigma^2 I$. Then, the covariates for which the estimated coefficients

Algorithm 2: Screening for iterative hard thresholding

Data: Screening variance ς^2; set of covariates $\{\mathbf{x}_j, j = 1, \ldots, m\}$
Result: New set of covariates $\{\mathbf{x}_l, l \in \mathcal{A}\}$

Define the screening prior as $p_L(\boldsymbol{\beta}) \sim \text{Laplace}(0, \varsigma^2 I)$
for $j = 1, \ldots, m$ **do**
  $\boldsymbol{\beta}^{(0)} \leftarrow \big(\Delta\beta_{j,L}(\mathbf{0})\big)\mathbf{e}_j$
The active set $\mathcal{A}$ is then constructed as $\mathcal{A} = \{j \,|\, \boldsymbol{\beta}_j^{(0)} \neq 0\}$

Algorithm 3: Iterative hard thresholding

Data: IHT$_x$; k; σ_β^2; ς^2; $\mathbf{y} = (y_1, \ldots, y_n)'$; $\{\mathbf{x}_j, j = 1, \ldots, m\}$

Result: $\hat{\boldsymbol{\beta}}$

if $IHT_x \in \{IHT_2, IHT_3\}$ **then**
  $\{\mathbf{x}_l, l \in \mathcal{A}\} \leftarrow$ Algorithm 2 (ς^2; $\{\mathbf{x}_j, j = 1, \ldots, m\}$)
else
  $\mathcal{A} = \{1, \ldots, m\}$
Define the prior for initialisation: $p_L(\boldsymbol{\beta}) \sim \text{Laplace}(0, \sigma_\beta^2 I)$
for $l \in \mathcal{A}$ **do**
  $\boldsymbol{\beta}^{(0)} \leftarrow \big(\Delta_{l,L}(\mathbf{0})\big)\mathbf{e}_l$
$\boldsymbol{\beta}^{(1)} \leftarrow H_k\big(\boldsymbol{\beta}^{(1)}\big)$; set iteration counter $q = 1$.
while *not converged* **do**
  for $l \in \mathcal{A}$ **do**
    $\sigma_{\beta,j}^2(q) \leftarrow \big(\beta_j^{(q)}\big)^2/(\log(n)/2)$
  Update the prior: $p_N(\boldsymbol{\beta}) \sim \text{Normal}(0, \Sigma_\beta)$, $\Sigma_\beta = \text{diag}(\sigma_{\beta,l}^2), \forall l \in \mathcal{A}$
  if $IHT_x \in \{IHT_1, IHT_3\}$ **then**
    $\boldsymbol{\beta}^{(q+1)} \leftarrow$ Algorithm 1 ($\boldsymbol{\beta}^{(q)}$; γ)
  else
    for $l \in \mathcal{A}$ **do**
      $\boldsymbol{\beta}^{(q+1)} \leftarrow \boldsymbol{\beta}^{(q)} + \big(\Delta_{l,N}(\boldsymbol{\beta}^{(q)})\big)\mathbf{e}_l$
    $\boldsymbol{\beta}^{(q+1)} \leftarrow H_k\big(\boldsymbol{\beta}^{(q+1)}\big)$
  $q \leftarrow q + 1$
$\hat{\boldsymbol{\beta}} \leftarrow \boldsymbol{\beta}^{(q)}$

are zero, are discarded. Since lasso should estimate the least important covariates as nonzero, these covariates are highly unlikely to be selected by IHT. This extension is primarily meant to achieve faster convergence since the number of covariates shrinks substantially. The procedure is described in Algorithm 2. Note that the input describes the set of covariates per covariate instead of per patient, i.e., the index is over $j = 1, \ldots, m$ instead of $i = 1, \ldots, n$. Naturally, the j-th covariate is defined as $\mathbf{x}_j = (x_{1j}, x_{2j}, \ldots, x_{nj})'$.

Lastly, let us combine the novel methodology into one algorithm. To refer to a specific combination of extensions, we introduce the notation IHT$_0$ for the baseline IHT algorithm without any

extensions, IHT_1 where the step-halving extension is implemented, IHT_2 where the screening step is included, and IHT_3 when both extensions are performed. The completed algorithm is presented in Algorithm 3.

4.5 Evaluation

The performance of the IHT algorithm is evaluated internally with a holdout set containing 25% of the data, where we consider the discrimination and the calibration. Then in external validation, we employ the estimated models to predict the probability of experiencing the medical outcome. The performance of the models in the external databases is evaluated based on discrimination and calibration, similarly to the internal evaluation.

For discrimination, we consider the concordance of the observed and predicted probabilities. For LR, this is similar to computing the area under the receiver operating characteristic (AUC), which is the most commonly reported statistic for discrimination in binary clinical predictions (Steyerberg, 2019). For Cox, this measure is simply called the C-Statistic, but it has the same interpretation as the AUC. Concordance represents the degree to which the observed and predicted probabilities "agree" with each other. For instance, given that the observed probability for patient i is higher than for patient j, the patients are concordant if the predicted probability for patient i is also higher than the predicted probability of patient j. Let the predicted probability be denoted as $\hat{y}_i$ and the observed probability as y_i, then the concordance is defined as $\Pr[\hat{y}_i > \hat{y}_j | y_i > y_j]$.

The AUC or C-Statistic can be interpreted as an experiment (Therneau & Watson, 2015). Suppose a predictive model is presented with two patients, where one experiences the medical outcome and the other does not. The model is tasked with predicting which of these two patients will experience the medical outcome. If you repeat this experiment one hundred times with a different pair of patients each time, then the AUC or C-Statistic represent the number of times the model made the correct prediction. In other words, the AUC and C-Statistic indicate the probability that the model is able to discriminate well between two patients who have different outcomes. This measure can take values between 0 and 1, but is often scaled by 100 when reported. If the AUC is equal to 0.5, the predictions resemble random guesses. Moreover, by conditioning on a pair

of patients with different outcomes, the AUC and C-Statistic are independent of the calibration of the predicted probabilities. Lastly, the statistics are reported with a 95% confidence interval. These intervals are computed asymptotically. Detailed descriptions of the computations are given in Appendix A.

Lastly, calibration checks whether the predicted risks correspond to the observed risks, which is commonly evaluated by means of calibration plots (Steyerberg, 2019). In these plots, the observed probabilities are plotted against the predicted probabilities. These observations are fitted with a loess curve (Austin & Steyerberg, 2014). This curve is presented in combination with the ideal calibration line: a straight line with the intercept in zero and a slope of one.

5 Simulation Study

We perform a simulation study to confirm the ability of the IHT algorithm to select the best subset of covariates and estimate their coefficients correctly. Doing so in a controlled setting allows us to assess the nuances of the variable selection properties. The study design is to a significant extent inspired by the simulation study in Kawaguchi et al. (2020).

The simulated data set consists of $n = 40\,000$ subjects and $m = 5\,000$ covariates. We investigate two settings for the true coefficient vector: (1) the "concentrated" setting where the true coefficient only has ten equally distinguished nonzero values, $\boldsymbol{\beta}_{0,\text{conc}} = (\mathbf{1}_5, -\mathbf{1}_5, \mathbf{0}_{m-10})'$; and (2) the "diffused" setting where the level of "nonzeroness" is more varied, $\boldsymbol{\beta}_{0,\text{diff}} = (\mathbf{1}_5, -\mathbf{1}_5, \mathbf{0.1}_2, \mathbf{0.6}_3, \mathbf{0}_{m-15})'$. By combining the two settings with different values of the IHT parameter k, we can simulate some potential real life settings.

The simulations are applied to LR models and Cox models, and we generate one set of covariates that is used for both models. Denoting the covariate matrix, $\mathbf{X}$, as $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)'$ with $\mathbf{x}_i = (x_{i1}, \ldots, x_{im})'$, x_{ij} represents covariate value j for subject i. We start by generating $Z_{ij} \sim N(0, 1)$.

Then, by defining x_{ij} as

$$x_{ij} = \begin{cases} 1 & \text{if } |z_{ij}| > 1.96, \\ 0 & \text{if } |z_{ij}| \leq 1.96, \end{cases} \tag{12}$$

we generate a covariate set where for each subject 5% of the covariate values are nonzero on average. This way, we are effectively mimicking typical patient profiles: only a few patients have a specific condition (Suchard et al., 2013).

Subsequently, the outcomes for the LR model are generated from a Bernoulli distribution with subject specific probability $p_i = (1 + \exp(-\mathbf{x}_i'\boldsymbol{\beta}_0))^{-1}$: $Y_i \sim \text{Bernoulli}(p_i)$. For the Cox model, we assume baseline hazard $h_0(t) = 1$ and generate the following random variables: event time t_i is generated from a Weibull distribution as $T_i \sim \text{Weibull}(1, \exp(-\mathbf{x}_i'\boldsymbol{\beta}_0))$, censoring control u_i is generated from a Uniform distribution as $U_i \sim \text{Unif}(0, 10)$, and lastly, censoring time c_i is generated from a Weibull distribution as $C_i \sim \text{Weibull}(1, u_i \cdot \exp(-\mathbf{x}_i'\boldsymbol{\beta}_0))$. The outcome y_i is then set as $y_i = \min(t_i, c_i)$. Using the true coefficient vector from the concentrated and diffused settings, the censoring rates are approx. 25% and 15%, respectively.

The simulation results are evaluated by the number of nonzero elements in the estimated coef-

Table 1. Variable selection evaluation for the concentrated setting, i.e., with $\|\boldsymbol{\beta}_0\|_0 = 10$.

	k	$\|\hat{\boldsymbol{\beta}}\|_0$	$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\|_2$	BIC
Logistic Regression	5	5	2.24	5.32×10^4
	10	10	0.14	5.14×10^4
	15	10	0.14	5.14×10^4
Cox Regression	5	5	2.26	5.73×10^5
	10	10	0.10	5.67×10^5
	15	10	0.10	5.67×10^5

Table 2. Coefficient estimates for the concentrated setting.

	k	intercept	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$	$\hat{\beta}_5$	$\hat{\beta}_6$	$\hat{\beta}_7$	$\hat{\beta}_8$	$\hat{\beta}_9$	$\hat{\beta}_{10}$	$\sum_{j=11}^{m} \lvert\hat{\beta}_j\rvert$
$\boldsymbol{\beta}_0$		0.2	1	1	1	1	1	-1	-1	-1	-1	-1	0
LR	5	0.23	1.01	0	0.94	0	0	0	0	-0.97	-1.01	-1.01	0
	10	0.19	1.05	0.98	0.99	0.94	0.94	-0.97	-0.95	-1.02	-1.04	-1.06	0
	15	0.19	1.05	0.98	0.99	0.94	0.94	-0.97	-0.95	-1.02	-1.04	-1.06	0
Cox	5	-	0	0	0.88	0.92	0.91	0	-0.80	0	0	-0.80	0
	10	-	0.95	0.99	1.01	1.01	1.05	-1.00	-1.02	-0.99	-0.93	-1.01	0
	15	-	0.95	0.99	1.01	1.01	1.05	-1.00	-1.02	-0.99	-0.93	-1.01	0

ficient vector $\|\hat{\boldsymbol{\beta}}\|_0$, the Euclidean distance between the true coefficient values and their estimated counterparts $\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\|_2$, and the BIC score $(-2\mathcal{L}(\hat{\boldsymbol{\beta}}) + \log(n)\|\hat{\boldsymbol{\beta}}\|_0)$, where $\|\hat{\boldsymbol{\beta}}\|_0$ is the number of parameters we are estimating. We start with the concentrated setting: there are ten nonzero coefficients that are all equally important and three possibilities for the relationship between k and the number of nonzero coefficients $\|\boldsymbol{\beta}_0\|_0$; k is either smaller than, equal to, or larger than $\|\boldsymbol{\beta}_0\|_0$. To simulate these possibilities, we run three simulations for both LR and Cox models with $k = 5$, $k = 10$, and $k = 15$. The results of the simulation are presented in Table 1. Naturally, when $k = 5$, the algorithm cannot select all ten nonzero covariates. This explains the higher deviation between the estimated and true coefficient vector, as well as the higher BIC score for both models. However, it is important to confirm that the algorithm did not falsely select any zero covariates. The coefficient estimates are written in Table 2, which verifies that IHT estimated the zero coefficients exclusively as zero.

Subsequently, when $k = 10$ the algorithm perfectly selects the ten nonzero coefficients for both models, and estimates them close to their true values. As a result, the deviation has shrunk close to zero and the BIC score declined as well. Then, moving to $k = 15$, the question is whether IHT keeps its correct selection of the ten nonzero covariates without adding any zero covariates. Clearly, that is not the case; the estimation remains unchanged when we move from $k = 10$ to $k = 15$, confirming that the algorithm only selects and estimates the nonzero coefficients. Accordingly, the deviation and BIC score are equal when comparing $k = 10$ and $k = 15$ for both models.

The second setting is called the diffused setting, where there is some difference in the level of coefficient magnitude. We take the true coefficient vector of the concentrated setting and change five zero coefficients to two values of 0.1 and three values of 0.6. Hence, we now have fifteen nonzero covariates, but of unequal importance. To assess the algorithm's behaviour, we run the simulations four times for both models with $k = 5$, $k = 10$, $k = 15$, and $k = 20$. First, let us consider the case where $k = 5$ or $k = 10$. Since the first ten covariates are the most important ones, we check if IHT selected a subset of these for $k = 5$ and the complete set when $k = 10$. The evaluating statistics are presented in Table 3, and the coefficient estimates are given in Table 4. It is easy to confirm that the algorithm performed desirably. When $k = 5$ the algorithm selected a subset of the ten

most important predictors and when $k = 10$ it selected all ten.

Secondly, we focus on the five covariates of lesser importance and investigate whether they will be selected when IHT allows enough "space" for them by setting $k = 15$ or $k = 20$. From Table 4 we first quickly note that no zero covariates were selected. Then, the three covariates with true coefficient value 0.6, β_{13}, β_{14}, and β_{15}, are selected and estimated close to their true value. However, the other two nonzero covariates with true coefficient value 0.1, β_{11} and β_{12}, do not seem to be important enough to be selected. We note two possible explanations for this. The algorithm employs ridge regularisation, which may shrink the coefficient values of β_{11} and β_{12} to an extent that IHT does not consider them to be significantly different than zero. The other possibility is

Table 3. Variable selection evaluation for the diffused setting, i.e., with $\|\boldsymbol{\beta}_0\|_0 = 15$.

	k	$\|\hat{\boldsymbol{\beta}}\|_0$	$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\|_2$	BIC
Logistic Regression	5	5	2.47	5.28×10^4
	10	10	1.06	5.11×10^4
	15	13	0.22	5.07×10^4
	20	13	0.22	5.07×10^4
Cox Regression	5	5	2.50	6.46×10^5
	10	10	1.05	6.40×10^5
	15	13	0.18	6.39×10^5
	20	13	0.18	6.39×10^5

Table 4. Coefficient estimates for the diffused setting.

	k	intercept	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$	$\hat{\beta}_5$	$\hat{\beta}_6$	$\hat{\beta}_7$	$\hat{\beta}_8$	$\hat{\beta}_9$	$\hat{\beta}_{10}$
$\boldsymbol{\beta}_0$		0.2	1	1	1	1	1	-1	-1	-1	-1	-1
LR	5	0.32	0.97	0.94	0	0	0	0	0	-0.96	-1.01	-1.01
	10	0.29	1.02	0.98	0.96	0.95	0.93	-0.97	-0.90	-1.01	-1.05	-1.06
	15	0.21	1.02	0.99	0.96	0.95	0.94	-0.98	-0.92	-1.02	-1.05	-1.07
	20	0.21	1.02	0.99	0.96	0.95	0.94	-0.98	-0.92	-1.02	-1.05	-1.07
Cox	5	-	0	0	0	0.91	0.89	-0.80	-0.78	0	0	-0.80
	10	-	0.93	0.98	0.98	1.00	1.01	-0.98	-0.99	-0.97	-0.95	-1.00
	15	-	0.95	1.00	1.00	1.03	1.03	-1.01	-1.03	-0.99	-0.97	-1.03
	20	-	0.95	1.00	1.00	1.03	1.03	-1.01	-1.03	-0.99	-0.97	-1.03

	k	$\hat{\beta}_{11}$	$\hat{\beta}_{12}$	$\hat{\beta}_{13}$	$\hat{\beta}_{14}$	$\hat{\beta}_{15}$	$\sum_{j=16}^{m} \lvert\hat{\beta}_j\rvert$
$\boldsymbol{\beta}_0$		0.1	0.1	0.6	0.6	0.6	0
LR	5	0	0	0	0	0	0
	10	0	0	0	0	0	0
	15	0	0	0.54	0.60	0.59	0
	20	0	0	0.54	0.60	0.59	0
Cox	5	0	0	0	0	0	0
	10	0	0	0	0	0	0
	15	0	0	0.60	0.54	0.57	0
	20	0	0	0.60	0.54	0.57	0

that the estimation converged in a local optimum. The latter seems unlikely, since that would mean the algorithm did not find the global optimum in all four cases. But, more simulations and more complex ones should be performed to rule out this possibility with certainty.

For this research, it is sufficient to be aware of this behaviour in the variable selection. However, it is interesting to discuss the implications. On the one hand, our efforts are aimed at finding parsimonious models with a manageable number of predictors. If there are covariates that may technically be nonzero, but with negligible impact, it is a positive result that they are not included. On the other hand, we clearly allowed space them by setting k at a certain value, and then we might wish all nonzero covariates to be selected, regardless of the covariate's impact.

Lastly, it may seem strange that the simulation study did not pay any attention to the extensions. First of all, the simulations were performed for all four possible combinations of extensions, and the extensions did not change the estimations. Secondly, it was a possibility to evaluate the estimation time and find if there were any efficiency gains for the extensions. However, that would require a more complicated simulation study, e.g., where the covariates are correlated. The efficiency gains will be addressed in the real data application, and since we had to set priorities, the decision was made to refrain from more complicated simulations. Nevertheless, this topic could be recommended to address in future research. With a more complicated simulation, it would also be interesting to evaluate the behaviour of lasso's variable selection compared to IHT. As to why this was not addressed in this research, the same argument applies. The IHT algorithm is extensively compared to the lasso algorithm in the real data application, and meaningful comparison in simulation requires a more complex simulation design.

6 Real Data Application

In this section, the methodology is applied to a real data set. We describe the clinical prediction problem, the specifications of the model, and then evaluate the performance of the novel methodology. First, we compare IHT to the benchmark method lasso, after which we evaluate the performance of the extensions, and lastly, we find out if the results hold in external validation.

The real data set considers patients with atrial fibrillation or flutter (AFF) and investigates whether this group of patients is at increased risk of all-cause mortality. Atrial fibrillation and atrial flutter are categorised as hearth rhythm disturbances. Vidaillet et al. (2002) provide evidence that there is increased risk of mortality for patients with AFF. They performed a study with 577 patients and 577 controls, and found that mortality among patients with AFF was nearly 7.8-fold higher at 6 months and 2.5-fold higher at the last follow-up. In a 2013 study, Andersson et al. found with Cox Regression that three concomitant diseases—neoplasm, chronic renal failure, and chronic obstructive pulmonary disease—were the most important predictors.

In our application, we select patients into the target cohort that have a diagnosis for atrial fibrillation or atrial flutter, or both. They enter the cohort only once, based on their first diagnosis. Furthermore, if the patient does not have at least 365 days or prior observation, this individual is excluded from the target cohort. Then, patients are included in the outcome cohort when they pass away within the time-at-risk, which is three years. Our analysis is run with data from the Integrated Primary Care Information (IPCI) database. More information on this database is included in Appendix B. The external validations are run in three different databases: CCAE, MDCD, and MDCR. The population size, outcome count, and observed risk for these four databases is presented in Table 5. The covariates that are included are (1) age; (2) ethnicity; (3) gender; (4) index month; and (5) race for demographic information, as well as the long and short term presence of (1) conditions; (2) devices exposure; (3) drug use; (4) measurements; (5) observations; and (6) procedures. A short term presence implies the presence is recorded in the last 30 days, whereas a long term presence means present in the last 365 days.

The hyperparameters selected for this application are shown in Table 6. In order to show the behaviour of lasso, we compare two lasso models with the various IHT models. The first lasso model is based on hyperparameters that were selected for optimal predictive performance. As

Table 5. Population size and outcome count for the four databases used.

Database	Population size	Outcome count	Observed risk (%)
IPCI	67,511	4,892	7.25
CCAE	1,204,068	22,045	1.83
MDCD	630,022	85,585	13.58
MDCR	1,601,565	80,964	5.06

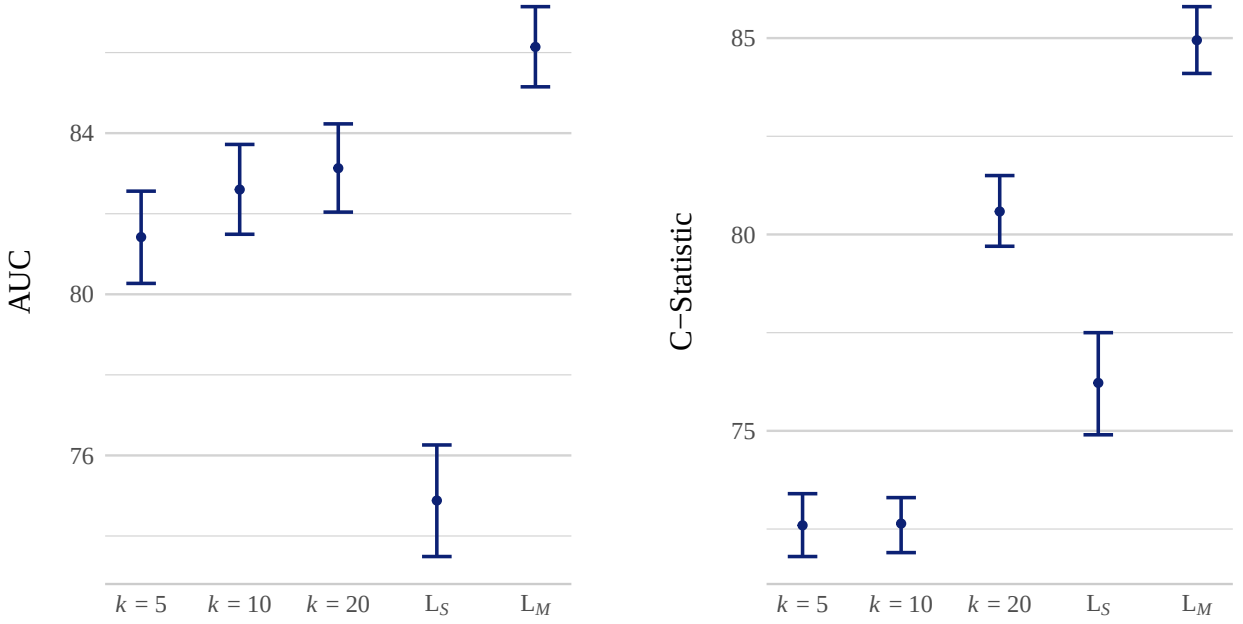
Table 6. Selected hyperparameters. The first row and the bottom three rows are selected based on predictive performance in training, whereas the second row of hyperparameters ($\sigma_\beta^2 = 10^{-5}$ and $\sigma_\beta^2 = 10^{-6}$) is manually set to offer another comparison to IHT.

Method		Logistic Regression		Cox Regression	
		σ_β^2	ς^2	σ_β^2	ς^2
Lasso	L_M	10^{-2}	-	10^{-2}	-
	L_S	10^{-5}	-	10^{-6}	-
IHT_0 & IHT_1	$k \in \{5, 10, 20\}$	10^{-2}	-	10^{-3}	-
IHT_2 & IHT_3	$k \in \{5, 20\}$	10^{-2}	10^{-3}	10^{-3}	10^{-4}
	$k \in \{10\}$	10^{-2}	10^{-4}	10^{-3}	10^{-4}

this requires a relatively mild lasso penalty, we denote this model as L_M. The full procedure is described in Appendix C, and for the first lasso model, resulted in the prior variance $\sigma_\beta^2 = 10^{-2}$ for both LR and Cox. The second lasso model aims to show the predictive performance when the penalty is strict to such an extent that the number of nonzero covariates is similar to the IHT parameter k, i.e., around ten nonzero covariates. Since the penalty for this lasso model is strong compared to the first model, we denote this model as L_S. Hence, the second set of variances for lasso ($\sigma_\beta^2 = 10^{-5}$ and $\sigma_\beta^2 = 10^{-6}$) were selected independent of predictive performance and are left out of the description in Appendix C. Remark that for IHT, the step-halving extension does not require any additional hyperparameters, and the use of step-halving also does not influence the hyperparameter selection. Therefore, the hyperparameters are the same between IHT_0 and IHT_1 (IHT, resp., without and with step-halving), and between IHT_2 and IHT_3 (IHT with screening and, resp., without and with step-halving). Further details on the hyperparameter selection for IHT are also included in Appendix C.

6.1 Iterative Hard Thresholding vs Lasso

Our first interest is to investigate the performance of IHT compared to the benchmark method lasso. For each IHT implementation, the algorithm is run three times, each with a different value for parameter k; we set $k = 5$, $k = 10$, or $k = 20$. The results for the LR and Cox model are shown in Figure 3. The mild lasso algorithm selected 355 nonzero covariates for LR and 423 for Cox, whereas the strong lasso algorithm selected 9 nonzero covariates for LR and 14 for Cox. Unsurprisingly, the L_M model achieves the highest performance. For the LR model, even though the drop in the



(a) Logistic Regression **(b)** Cox Regression

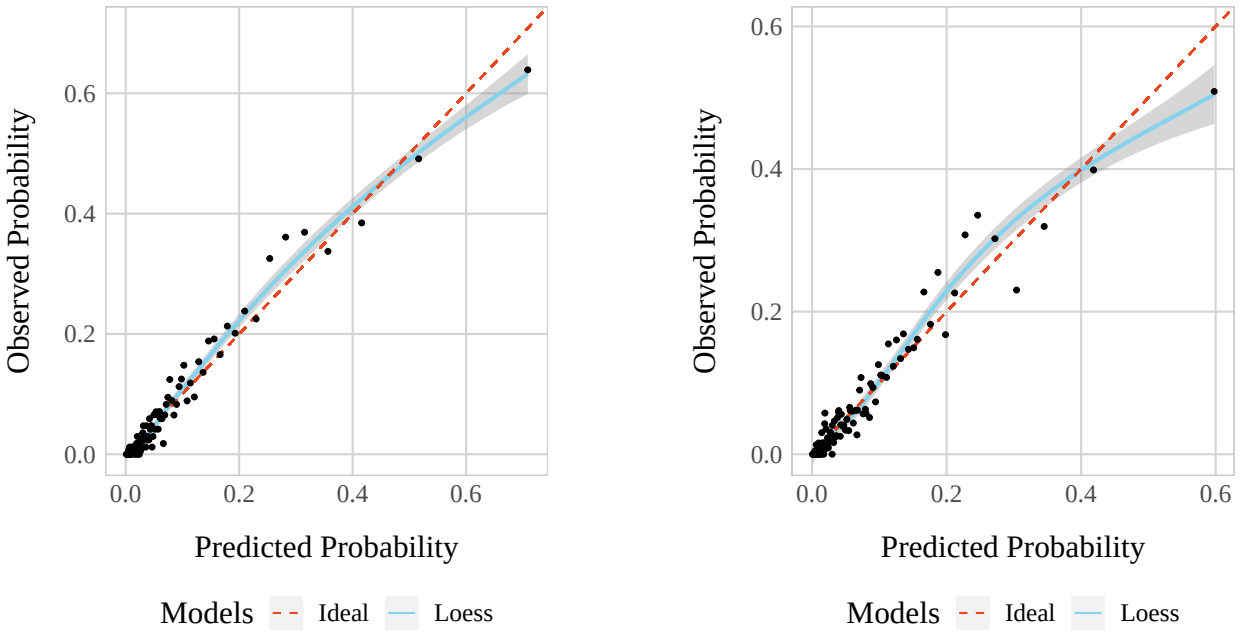
Figure 3. Performance of Lasso and IHT.

AUC is significant, the performance of the IHT models is still satisfactory. Especially the result for the IHT model with $k = 10$ is positive, since ten is a very manageable number of predictors to work with in practice. In other words, the slight drop in performance is a welcome compromise. Furthermore, if we compare the performance of the IHT models with the strong lasso algorithm, L_S, which selects nine nonzero covariates, we find that IHT substantially outperforms lasso.

Then, in Figure 3b we see that the drop of IHT compared to L_M is more severe for the Cox models, although less so when you allow for a larger k. Where LR models only try to predict the presence of a medical event, Cox models try to predict the timing of this event. It does not surprise then that the model would need (a few) more covariates in order to predict that well. Secondly, the strong lasso algorithm seems to achieve comparable performance to IHT. The C-Statistic with L_S is roughly halfway between the $k = 10$ and the $k = 20$ IHT models, where the L_S selects 14 nonzero covariates.

We also briefly address the calibration of the models. In Figure 4, the calibration of the mild lasso model and the IHT model with $k = 10$ is plotted[3]. As mentioned before, the data is well

[3]The calibration plots are only shown for the LR models. The Cox models are used much less frequently, hence



(a) Lasso with $\sigma_\beta^2 = 10^{-2}$ (L_S) **(b)** IHT with $k = 10$

Figure 4. Calibration plots of two Logistic Regression models.

calibrated if the predicted risk is similar to the observed risk. In other words, the intercept should be close to zero and the slope close to one. It is easy to confirm that the data is well calibrated.

6.2 Iterative Hard Thresholding Extensions

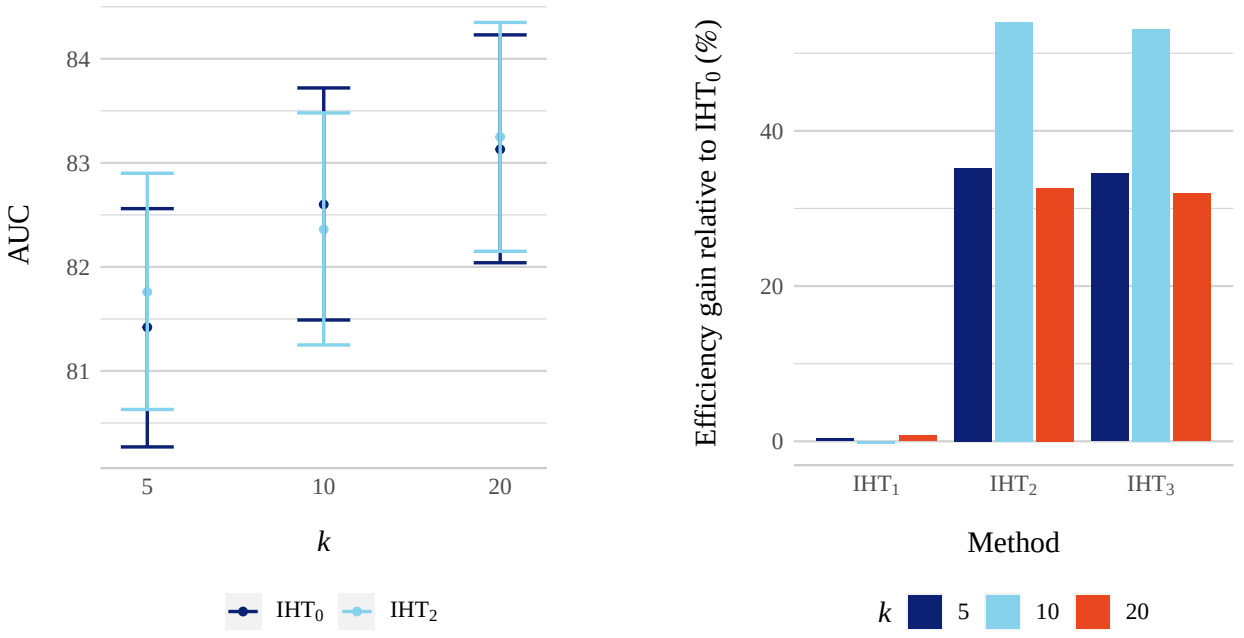
Our next focus is on the performance of the methodology extensions, namely step-halving and screening. As mentioned before, we refer to the IHT model without any extensions as IHT_0, with step-halving as IHT_1, with screening as IHT_2, and with both extensions as IHT_3. Also, again, we run all models three times with different values for the IHT parameter k, namely with $k = 5$, $k = 10$, and $k = 20$. In Figure 5, the results for the LR models are displayed, with the predictive performance in Figure 5a and the efficiency gains in Figure 5b. The efficiency gains are given in percentages for IHT_1, IHT_2, and IHT_3 relative to IHT_0. The estimation time of the IHT_0 models was generally eight minutes.

Most notable is the fact that Figure 5a only shows the results for IHT_0 and IHT_2. That is because step-halving had no impact on the location that the algorithm converges in. In other words,

the calibration evaluation is currently not included in the PLP software.

every single coefficient estimate was the same regardless of whether step-halving was performed. Screening, on the other hand, did have an impact, although insignificant. Note that this is not a negative result in the slightest. The extensions were primarily designed to achieve faster estimation times. In Figure 5b we find that the contribution of step-halving to shrinking the estimation time is nonexistent, since there is no efficiency gain when just step-halving is applied. The efficiency gains also remain constant when step-halving is added to IHT with screening (IHT_2). However, the screening extension results in substantial efficiency gains, with at its peak cutting the estimation time in half (for $k = 10$).

Then, let us evaluate the same results for the Cox models, which are shown in Figure 6. The severe drop in performance for $k = 5$ and $k = 10$ seems to have been recovered by the screening step. This is a great result but it is not necessarily expected. The screening step discards any covariates that are highly unlikely to end up in the best subset. Without these covariates the algorithm might be less disturbed by them, and make it easier to find a better subset. Alternatively, the reason could be in the IHT specification without any extensions. It may also just be a coincidence or related to some other yet undetected cause, and calls for further investigation in future research. Nevertheless,

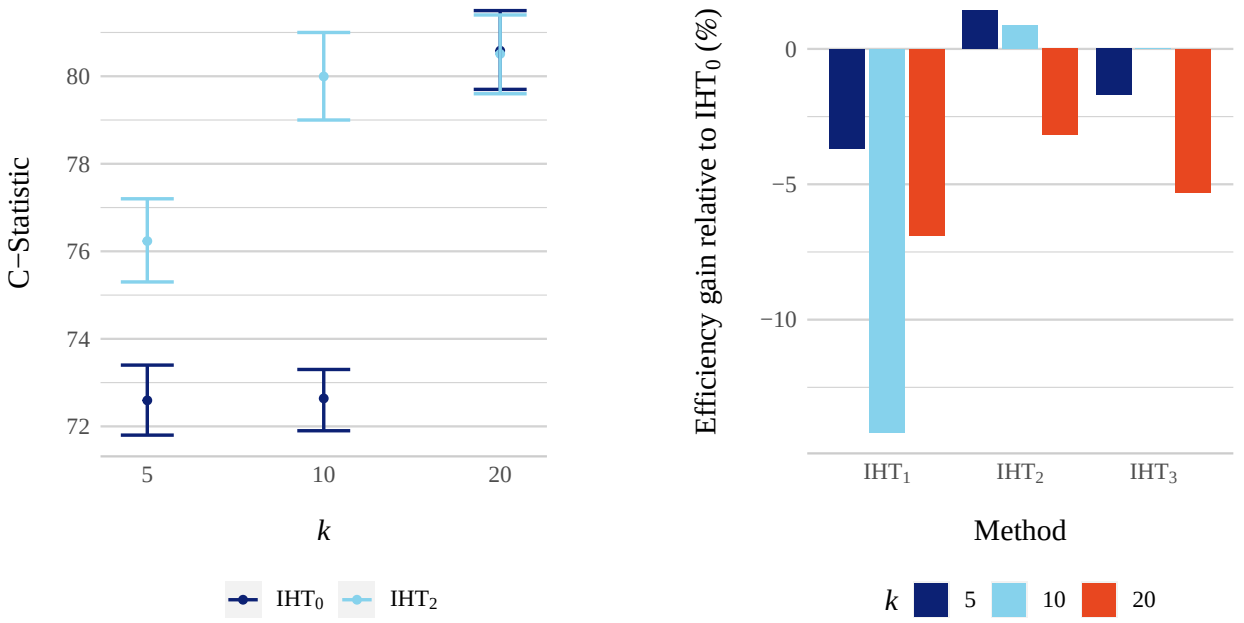


(a) Predictive performance **(b)** Efficiency gains

Figure 5. Evaluation of extensions for Logistic Regression.

it is promising result. Note that the performance for $k = 20$ does not differ significantly. Moreover, it should be noted that IHT outperforms the strong lasso if we combine the $k = 5$ and $k = 10$ results of the IHT_2 algorithm with the $k = 20$ result of the IHT_0 algorithm. The C-Statistic for the L_S model is 76.2, which is equal to the lowest C-Statistic of the combined the IHT models.

The step-halving extension seems to worsen the overall performance by increasing the estimation time. Even though this is not the intended result, there is an explanation for it. The step-halving extension introduces additional computations before allowing the algorithm to move to a new cycle of cyclic coordinate descent. If the time to perform these additional computations do not way up against the time gain they provide, the algorithm will be slower. Then for the screening step we see some interesting results. The extension seems to introduce small efficiency gains for $k = 5$ and $k = 10$, while also improving the predictive performance. The extension does not seem to add much value for $k = 20$; the predictive performance does not change significantly and the estimation time has increased slightly.



(a) Predictive performance **(b)** Efficiency gains

Figure 6. Evaluation of extensions for Cox Regression.

6.3 Selected Covariate Subsets

Having seen the overall performance of the algorithm with all of its possible extensions, let us address which covariates were actually selected. The selected covariates are listed in Table 7, including their estimates for the IHT_0 and IHT_2 algorithm with $k = 10$ for both LR and Cox. The same tables for the algorithms with $k = 5$ and $k = 20$ are included in Appendix D.

Firstly, let us address a question that is not included in the tables but still important to answer. The lasso algorithm is used as a benchmark to compare with IHT and for the screening extension. For the latter, it is interesting to investigate if the set of nonzero covariates selected by lasso leaves out any covariates that are selected by the IHT algorithm that did not use the screening step. Namely, if lasso leaves out any of those covariates, that could be an indication that lasso is not a suitable method for the screening step, since the screening step should only discard covariates that are highly unlikely to be selected as nonzero. The list of covariates that lasso estimates as nonzero is too extensive to display in this paper. However, for our application, it has been confirmed that

Table 7. Covariate selection and coefficient estimation for the IHT model with $k = 10$.

Domain	Covariate		LR		Cox	
			IHT_0	IHT_2	IHT_0	IHT_2
Demographic	Age		9.591	8.265	9.147	8.214
	Female				-0.274	-0.385
Conditions	Malignant neoplastic disease	(short term)	0.972		0.777	0.703
	Malignant neoplastic disease	(long term)		1.281		0.789
	Primary malignant neoplasm	(long term)			1.081	
	Primary malignant neoplasm of trunk	(long term)	1.507			
	Ulcer of skin or mucosa	(long term)			0.671	
Drugs	Antiemetics and antinauseants	(long term)	1.659			
	Antiiflammatory and antirheumatic	(long term)		-0.258		-0.217
	For alimentary tract and metabolism	(long term)			0.518	
	For obstructive airway diseases	(immediate)		0.426		
	Prednisolone	(long term)		0.473		
	Opioids	(long term)		0.557		0.579
	High-ceiling diuretics	(long term)		0.636		0.583
	Antipsychotics	(long term)			0.740	
	Vitamin A and/or D	(long term)				0.478
	All other theurapeutic products	(long term)	1.639			
Measurements	Oxygen saturation in arterial blood	(short term)	0.756	0.598	0.719	0.701
	Glomerular filtration rate	(long term)	-0.369			
Other	Charlson index - Romano adaptation*		2.909		2.244	
	CHADS2**			0.743		0.693

* The Charlson index is a comorbidity index

** CHADS2 carries information about the risk of experiencing a stroke

the list of covariates selected by lasso does not leave out any covariates that are selected by the IHT algorithms that do not include the screening step, which indicates that the lasso algorithm is an appropriate method for the screening step.

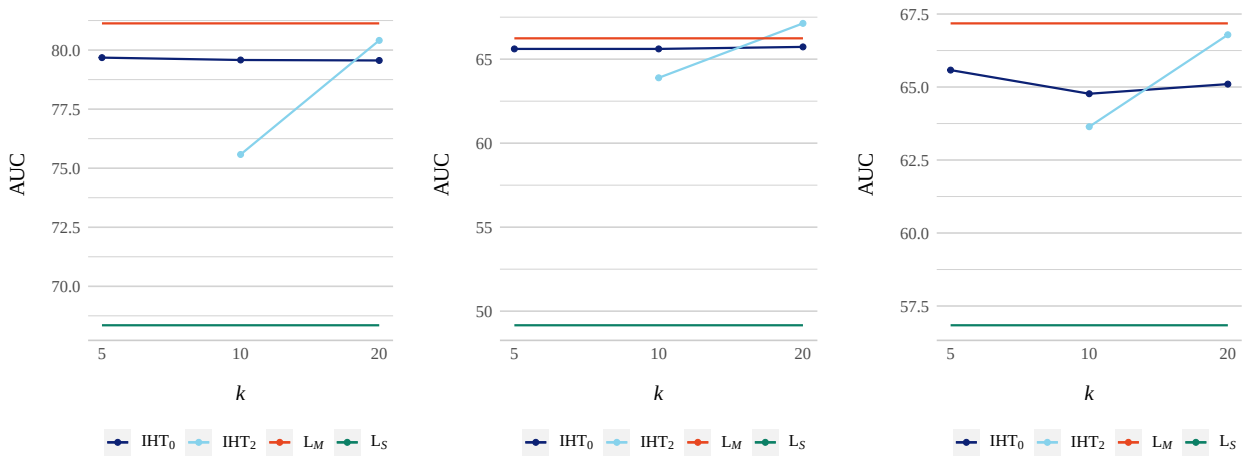
Then, reviewing the selections of covariates in Table 7, it is clear that these selections differ between the various IHT algorithms, but the covariates share some similarities. For instance, neoplasm was found to be an important predictor for all-cause mortality for patients with AFF (Andersson et al., 2013), and all sets of selected covariate contain a covariate that carries information about the presence of neoplasm. Age, unsurprisingly, seems to be an important predictor as well.

Secondly, we investigate which covariates are selected when only the value for k changes but the algorithm remains the same. We address the following question: When moving to a larger k, while all other specifications are kept equal, does the selection of covariates change in such a way that covariates that were included before and not included with the larger k? Or, the other way around, when decreasing k, does IHT only select less covariates or also other covariates? The short answer is, most of the time, yes it does only select less covariates from the same set that was previously selected. In our case specifically, the answer is yes for IHT_0 in LR context and IHT_2 in Cox models. However, for the other two settings, there are a one or two covariates that are included for the $k = 5$ specification but not for the $k = 10$ or the $k = 20$ specifications. In conclusion, increasing k generally means adding covariates without removing any.

6.4 External Validation

One of the most important and interesting unknowns before the execution of the research was whether the results from IHT would hold up in external validation. The CDM managed by the OHDSI community makes external validation possible for every single PLP model, as long as the database is linked to the CDM. The models in this research have been run in three external databases: CCAE, MDCD, and MDCR. Their population sizes and outcome counts are presented in Table 5. More information about these databases is included in Appendix B.

To see if the database is generally suited for predicting our clinical prediction problem, we run the mild lasso algorithm. Additionally, we run the IHT_0, IHT_2, and strong lasso algorithms. Note



(a) CCAE **(b)** MDCD **(c)** MDCR

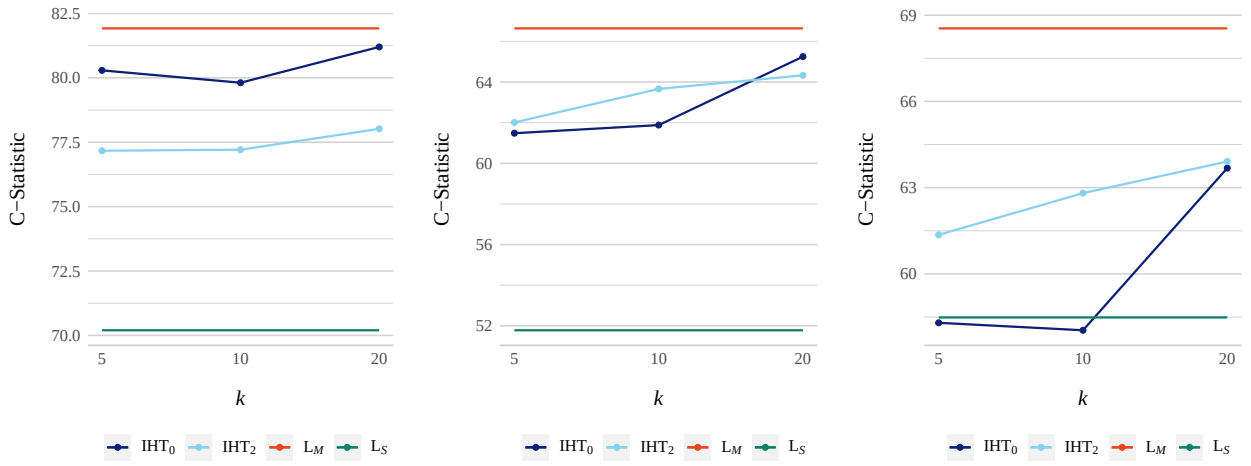
Figure 7. External validation performance for Logistic Regression.

that IHT_1 and IHT_3 are excluded, since these estimated models are identical to, resp., IHT_0 and IHT_2. The external validation results offer two perspectives. Firstly, we find whether the predictive performance for this specific clinical prediction problem holds in external validation. Secondly, we compare the two lasso models with the IHT models in each database. For our research, the latter is more important, although the first perspective is still relevant. Namely, the mild lasso model gives an indication of the maximum performance that IHT can achieve. In other words, if L_M performs poorly in external validation, than the severity of the drop in performance for the IHT models is less meaningful than when L_M performs well.

The results for the LR and Cox models are shown in Figures 7 and 8, respectively[4]. First of all, we note that the performance in CCAE is much better than in the other two databases. The predictive performance of the mild lasso in this database is similar to the predictive performance in IPCI, which cannot be said for the other two databases. Hence, the CCAE database is most relevant for our evaluation.

In Figure 7a, we find that IHT_0 achieves consistent performance across different values of k, and the AUC is only slightly below L_M. The performance for IHT_0 decreases more substantially for $k = 10$, but recovers for $k = 20$. The performance of L_S, on the other hand, is significantly below that of IHT. Subsequently, the drop performance of IHT in the other two databases is barely

[4]The IHT_2 model with $k = 5$ for LR did not run in any external database. This was due to some storage error and not related to the IHT algorithm.

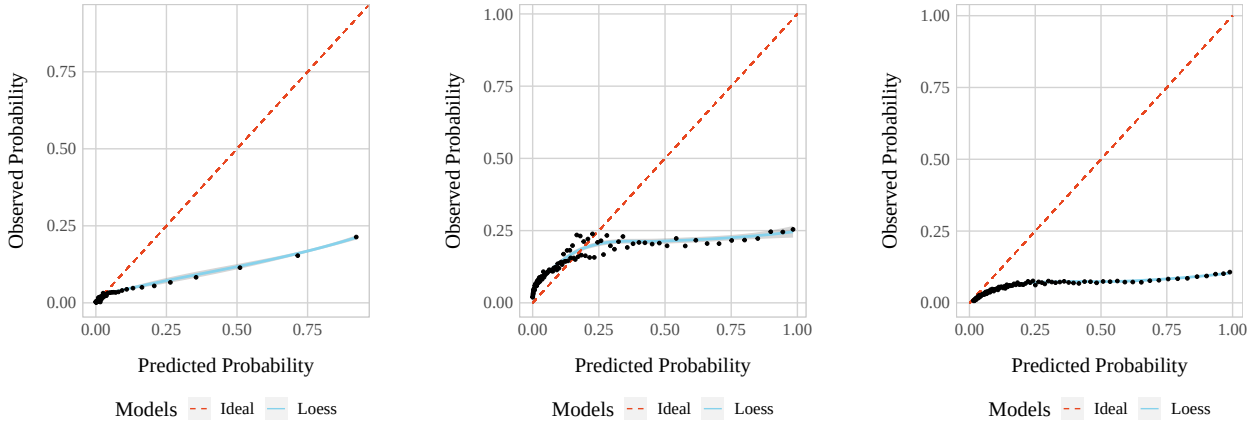


(a) CCAE **(b)** MDCD **(c)** MDCR

Figure 8. External validation performance for Cox Regression.

significant compared to the mild lasso model. Again, we find that the strong lasso model performs very poorly. However, since the performance of L_M in itself is quite poor, there is less space for IHT to perform worse, hence why these databases are not as relevant. Then in Figure 8, we find similar results in CCAE and MDCD for the Cox models compared to LR. IHT is capable of retaining its predictive performance in external validation on a comparable level with the mild lasso model, while the strong lasso model performs significantly worse. However, in the MDCR database, the performance of the IHT models is quite poor.

Lastly, we consider the calibration plots for the three databases. The calibration for IHT_0 with $k = 10$ is shown in Figure 9. We see that, while the intercept is close to zero, the slope is not close to



(a) CCAE **(b)** MDCD **(c)** MDCR

Figure 9. Calibration plots for the three external databases for the IHT_0 algorithm with $k = 10$.

one. The discrimination results are still valid, but the predicted probabilities should be recalibrated before the models are implemented in practice (Steyerberg, 2019). Overall, the performance of IHT is excellent, and this seems to confirm that the methodology can hold in external validation.

7 Conclusion

The primary goal of this research was to develop a novel method that finds parsimonious PLP models by leveraging the inevitable sparsity found in PLP data. PLP models are clinical prediction models that predict the risk of some medical outcome based on patient-level information. The methodology builds upon the concept of iterative hard thresholding. In IHT, we set a hard threshold, i.e., we allow the algorithm to select at most k covariates. The algorithm then estimates the coefficients in an iterative manner, where only the k largest coefficient estimates are retained at each iteration; the other are set to zero. The methodology has been implemented for Logistic Regressions (LRs) and Cox models, but can be applied to generalized linear models as well.

Firstly, we altered the current IHT methodology such that it can manage the high-dimensionality and sparsity of PLP data. This is achieved by implementing cyclic coordinate descent instead of gradient descent and introducing an ℓ_2-norm penalty to the log-likelihood that is optimised by CCD. This procedure is initialised by a warm start, i.e., a lasso estimation. The variable selection property of the algorithm was verified in a simulation study, and we found that this IHT algorithm outperforms the lasso model if both models select a similar number of nonzero covariates. If the lasso algorithm is optimised with respect to predictive performance, IHT achieve a predictive performance that is only slightly below lasso's performance, but IHT then bases its prediction on approximately ten covariates whereas lasso uses a few hundred covariates.

Secondly, we introduced two extensions, step-halving and screening, which were intended to decrease the estimation time, i.e., provide gains in efficiency. Step-halving attempts to force a descent in the negative log-likelihood at each iteration. Screening shrinks the set of covariates by performing a lasso-regularised regression and excluding covariates for which the coefficient is estimated as zero. The step-halving extension did not have any impact on the selected covariates,

nor on their estimated coefficient values. Moreover, the extension was unsuccessful in providing any efficiency gains, and at times even increased the estimation time. The screening extension, on the other hand, was successful and provided significant efficiency gains. A positively surprising result was the fact that screening improved the predictive performance of the IHT algorithm for the Cox models.

Thirdly, with this research, it was the first time that an automated feature selection algorithm was evaluated in external validation for clinical prediction modeling. We found that the performance of IHT carries over to external databases, given that the database is suited for the specific clinical prediction problem. This condition was assessed by applying the optimised lasso models to the databases, and their performance provided a benchmark for the IHT algorithm. IHT managed to stay close to the benchmark in predictive performance and outperform lasso when lasso is tasked to select a similar number of covariates.

In conclusion, the novel IHT algorithm showed the ability to select some of the most important predictors, while simultaneously estimating their coefficients, and attained good predictive performance when applied to the holdout set as well as during external validation. This means that the algorithm provides a competitive and scalable approach to automated feature selection in PLP models, which in itself is useful for the implementation of predictive modeling in clinical practice.

The research was limited most prominently by the absence of a full cross-validation procedure for the IHT algorithm. Designing such a procedure does not pose major complications, but the implementation was infeasible for this research. A detailed explanation for this is provided in Appendix C. As a result, the selected hyperparameters may not have been optimal. This does not invalidate our assessment of the IHT algorithm. IHT outperformed lasso, and optimising the hyperparameters could only improve IHT's performance. However, this limitation has implications for the assessment of the proposed extensions. Particularly the screening extension, which performed surprisingly well, should be evaluated with the optimal hyperparameters. The screening step is influenced by the prior variance and it should be investigated how to optimise this prior variance. Hence, we recommend the implementation of a cross-validation procedure for future research, which should enable a more detailed evaluation of the screening extension.

Lastly, the real data application showed the performance of IHT in one clinical prediction problem. The algorithm was run with different settings, such as the varying levels for the IHT threshold k, and for two different model specifications. Nevertheless, to assess the performance of IHT in more general terms, the procedures should be applied to more PLP problems, which we also recommend for future research. The application to multiple problems was not the focus of this work. Hence, it is not necessarily a limitation of the research, but we should be careful in formalising our judgement of the performance of IHT. This work derived the algorithm and the real data application showed promising results. Thereby, we can conclude with certainty that the novel algorithm holds the potential to become an important player in the implementation of predictive modeling in clinical practice.

References

Andersson, T., Magnuson, A., Bryngelsson, I.-L., Frøbert, O., Henriksson, K. M., Edvardsson, N., & Poçi, D. (2013). All-cause mortality in 272 186 patients hospitalized with incident atrial fibrillation 1995–2008: a swedish nationwide long-term case–control study. *European heart journal*, *34*(14), 1061–1067.

Austin, P. C., & Steyerberg, E. W. (2014). Graphical assessment of internal and external calibration of logistic regression models by using loess smoothers. *Statistics in medicine*, *33*(3), 517–535.

Bandeira, A. S., Dobriban, E., Mixon, D. G., & Sawin, W. F. (2013). Certifying the restricted isometry property is hard. *IEEE transactions on information theory*, *59*(6), 3448–3450.

Bertsimas, D., King, A., & Mazumder, R. (2016). Best subset selection via a modern optimization lens. *Annals of statistics*, *44*(2), 813–852.

Bertsimas, D., & Van Parys, B. (2020). Sparse high-dimensional regression: Exact scalable algorithms and phase transitions. *The Annals of Statistics*, *48*(1), 300–323.

Blumensath, T. (2012). Accelerated iterative hard thresholding. *Signal Processing*, *92*(3), 752–756.

Blumensath, T., & Davies, M. E. (2009). Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, *27*(3), 265–274.

Blumensath, T., & Davies, M. E. (2010). Normalized iterative hard thresholding: Guaranteed stability and performance. *IEEE Journal of selected topics in signal processing*, *4*(2), 298–309.

Chen, X., Liu, C. C., & Xu, S. (2021). An efficient algorithm for joint feature screening in ultrahigh-dimensional cox's model. *Computational Statistics*, *36*(2), 885–910.

Chen, Y., & Zhao, Y. (2021). Efficient sparse estimation on interval-censored data with approximated l0 norm: Application to child mortality. *PloS one*, *16*(4), e0249359.

Chu, B. B., Keys, K. L., German, C. A., Zhou, H., Zhou, J. J., Sobel, E. M., ... Lange, K. (2020). Iterative hard thresholding in genome-wide association studies: Generalized linear models, prior weights, and double sparsity. *GigaScience*, *9*(6), giaa044.

Erion, G., Janizek, J. D., Hudelson, C., Utarnachitt, R. B., McCoy, A. M., Sayre, M. R., ... Lee,

S.-I. (2021). Coai: Cost-aware artificial intelligence for health care. *medRxiv*.

Fan, J., Gong, W., & Sun, Q. (2021). A provable two-stage algorithm for penalized hazards regression. *arXiv preprint arXiv:2107.02730*.

Hazimeh, H., & Mazumder, R. (2020). Fast best subset selection: Coordinate descent and local combinatorial optimization algorithms. *Operations Research*, *68*(5), 1517–1537.

Herrity, K. K., Gilbert, A. C., & Tropp, J. A. (2006). Sparse approximation via iterative thresholding. In *2006 ieee international conference on acoustics speech and signal processing proceedings* (Vol. 3, pp. III–III).

Kawaguchi, E. S., Suchard, M. A., Liu, Z., & Li, G. (2020). A surrogate ℓ_0 sparse cox's regression with applications to sparse high-dimensional massive sample size time-to-event data. *Statistics in medicine*, *39*(6), 675–686.

Li, X., Xie, S., Zeng, D., & Wang, Y. (2018). Efficient ℓ_0-norm feature selection based on augmented and penalized minimization. *Statistics in medicine*, *37*(3), 473–486.

Liu, Y., Xu, J., & Li, G. (2021). Sure joint feature screening in nonparametric transformation model for right censored data. *Canadian Journal of Statistics*, *49*(2), 549–565.

Liu, Z., Sun, F., & McGovern, D. P. (2017). Sparse generalized linear model with l_0 approximation for feature selection and prediction with big omics data. *BioData mining*, *10*(1), 1–12.

Liu, Z., & Xiong, Z. (2022). Non-marginal feature screening for additive hazard model with ultrahigh-dimensional covariates. *Communications in Statistics-Theory and Methods*, *51*(6), 1876–1894.

Mittal, S., Madigan, D., Burd, R. S., & Suchard, M. A. (2014). High-dimensional, massive sample-size cox proportional hazards regression for survival analysis. *Biostatistics*, *15*(2), 207–221.

Murdoch, T. B., & Detsky, A. S. (2013, 04). The inevitable application of big data to health care. *JAMA*, *309*(13), 1351-1352.

OHDSI. (2021). *The book of ohdsi: Observational health data sciences and informatics*. Author. Retrieved from https://ohdsi.github.io/TheBookOfOhdsi/

Reps, J. M., Schuemie, M. J., Suchard, M. A., Ryan, P. B., & Rijnbeek, P. R. (2018). Design and implementation of a standardized framework to generate and evaluate patient-level prediction

models using observational healthcare data. *Journal of the American Medical Informatics Association*, *25*(8), 969–975.

Steyerberg, E. W. (2019). *Clinical prediction models*. Springer, Cham.

Su, X., Wijayasinghe, C. S., Fan, J., & Zhang, Y. (2016). Sparse estimation of cox proportional hazards models via approximated information criteria. *Biometrics*, *72*(3), 751–759.

Suchard, M. A., Simpson, S. E., Zorych, I., Ryan, P., & Madigan, D. (2013). Massive parallelization of serial inference algorithms for a complex generalized linear model. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, *23*(1), 1–17.

Therneau, T. M., & Watson, D. A. (2015). *Technical report series no. 85: The concordance statistic and the cox model* (Tech. Rep.). Mayo Clinic: Department of Health Science Research.

Topol, E. J. (2015). *The patient will see you now: the future of medicine is in your hands*. Basic Books New York.

Vidaillet, H., Granada, J. F., Chyou, P.-H., Maassen, K., Ortiz, M., Pulido, J. N., ... Hayes, J. (2002). A population-based study of mortality among patients with atrial fibrillation or flutter. *The American journal of medicine*, *113*(5), 365–370.

Xu, C., & Chen, J. (2014). The sparse mle for ultrahigh-dimensional feature screening. *Journal of the American Statistical Association*, *109*(507), 1257–1269.

Yang, G., Yu, Y., Li, R., & Buu, A. (2016). Feature screening in ultrahigh dimensional cox's model. *Statistica Sinica*, *26*, 881.

Zheng, Z., Fan, Y., & Lv, J. (2014). High dimensional thresholded regression and shrinkage effect. *Journal of the Royal Statistical Society: Series B: Statistical Methodology*, 627–649.

Zuo, Y., Stewart, T. G., & Blume, J. D. (2021). Variable selection in glm and cox models with second-generation p-values. *arXiv preprint arXiv:2109.09851*.

Appendix A Computation of the AUC and the C-Statistic

Here we outline the details regarding the computation of the AUC and the C-Statistic.

A.1 Computation of the AUC

First, we consider the AUC, which is computed within the PLP software (https://github.com/OHDSI/PatientLevelPrediction). First, we define the S operator, which is referred to as the Mann-Whitney kernel in the software, because the operator is identical to the one used for the Mann-Whitney U-Statistic:

$$S(x,y) = \begin{cases} 1 & \text{if } x > y, \\ \frac{1}{2} & \text{if } x = y, \\ 0 & \text{if } x < y\,. \end{cases} \tag{A.13}$$

Secondly, assume we have a set of n observed probabilities $y_i, i = 1, \ldots, n$. Since the AUC is computed with a binary outcome variable, $y_i = 0$ or $y_i = 1$. We compare each patient j who does not experience the medical outcome with each patient i who does experience the medical outcome, because we know for this patient pair (i, j) that $y_i > y_j$, namely $y_i = 1$ and $y_j = 0$. Then, we denote the predicted probability of patient i as $\hat{y}_i$. If $\hat{y}_i > \hat{y}_j$, the the pair (i, j) is concordant. The AUC is the average concordance of all patient pairs:

$$\text{AUC} = \frac{1}{n} \sum_{i:y_i=1} \sum_{j:y_j=0} S(\hat{y}_i, \hat{y}_j)\,. \tag{A.14}$$

To compute the 95% confidence interval of the AUC, we calculate the variance. First, we distinguish the number of cases (patients who experience the medical outcome), n_1, from the number of controls (patients who do not experience the medical outcome), n_0, such that $n = n_1 + n_0$. Secondly, we define two vectors, $\mathbf{a}_0$ and $\mathbf{a}_1$, that contain the averages of the S operator over the

controls and cases, respectively, as,

$$a_{0,j} = \frac{1}{n_1} \sum_{i:y_i=1} S(\hat{y}_i, \hat{y}_j)\,, \quad \forall j : y_j = 0, \tag{A.15}$$

$$a_{1,i} = \frac{1}{n_0} \sum_{j:y_j=0} S(\hat{y}_i, \hat{y}_j)\,, \quad \forall i : y_i = 1\,. \tag{A.16}$$

Subsequently, the variances of these vectors are calculated as

$$v_0 = \frac{1}{n_0 - 1} \sum_{j:y_j=0} (a_{0,j} - \text{AUC})^2, \tag{A.17}$$

$$v_1 = \frac{1}{n_1 - 1} \sum_{i:y_i=1} (a_{1,i} - \text{AUC})^2\,, \quad \forall i : y_i = 1\,. \tag{A.18}$$

Lastly, with the variance of the AUC defined as $v = v_0/n_0 + v_1/n_1$, the 95% confidence interval is asymptotically constructed as $\text{AUC} \pm 1.95\sqrt{v}$.

A.2 Computation of the C-Statistic

The C-Statistic is calculated by the concordance function of the survival package (see `https://github.com/therneau/survival/blob/master/R/concordance.R`). For the precise calculations of the C-Statistic we refer to the concordance vignette and Therneau and Watson (2015). However, we address two difficulties that arise specifically for survival models: how to handle ties of event times and how to handle censored data. It is rather straightforward. Namely, the C-Statistic only considers patient pair (i, j) if it is known that patient i experiences the medical outcome sooner than patient j. In other words, patients with tied observed event times are excluded, and patients who cannot be compared due to censoring are excluded (Therneau & Watson, 2015). For example, if patient i is right-censored at 10 months and patient j experiences the medical outcome at 13 months, it is uncertain if patient i experiences the medical outcome before patient j and the pair (i, j) is excluded.

Then, for all considered pairs (i, j), where it is known that patient i passed away before patient j —the medical outcome in our real data application is all-cause mortality, their contribution to

the C-Statistic is determined by the S operator in (A.14). The variance is estimated with the infinitesimal jackknife (Therneau & Watson, 2015), which is used to construct the 95% confidence interval in a similar manner as for the AUC.

Lastly, the AUC and the C-Statistic are independent of the calibration of the estimated probabilities. However, the C-Statistic is sensitive to the censoring pattern. See Chapter 15.2.11 of Steyerberg (2019) for potential alternatives.

Appendix B Data Background

The main database used in this research is the IPCI database. The IPCI database is managed by the Medical Informatics department of the Erasmus Medical Centre and consists of information that a general practitioner (GP) routinely records. It contains patient records from GP's in The Netherlands, including diagnoses, medication, and laboratory measurements. The data is anonymised and sent to the database twice per year. The first records date from 1996 and the most recent ones are from 2021. Information from 1.5 million patients has been collected by roughly 650 GP's in the last four years. For more information we refer to the IPCI website: https://www.ipci.nl/index.php.

For the external validation, we consider three external databases: Commercial Claims and Encounters (CCAE), Medicare Supplemental and Coordination of Benefits (MDCR), and Multi-State Medicaid Database (MDCD), which are all IBM MarketScan® databases. They are managed by Janssen, a pharmaceutical company that works closely together with the researchers from Erasmus MC (and the OHDSI network) on the development of the PLP methodology and software, as well as the common data model (CDM). The databases are linked to the CDM, and contain the patient profiles of various patients, who differ across databases based on their healthcare plan. For more information, see https://ohdsi.github.io/ETL-LambdaBuilder/docs/IBM_CCAE_MDCR and https://ohdsi.github.io/ETL-LambdaBuilder/docs/IBM_MDCD.

Appendix C Hyperparameter Selection

For the hyperparameter selection, the ideal approach would be cross-validation. If there had been unlimited availability of time for this research, we would have designed and tested a complete cross-validation procedure for IHT and the extensions in the PLP software (see https://github.com/OHDSI/PatientLevelPrediction). The decision was made not to do so, since this would be a rather complex process. For instance, the screening extension requires a variance parameter for the screening prior. The screening step is the first one in the algorithm. For us to evaluate the effect of a screening variance, the complete IHT algorithm should be executed, which is written in another package. Furthermore, the current cross-validation is also more complicated than a standard n-fold cross-validation; it consists of an automated grid-search (for more details see Suchard et al. 2013), and is controlled with C++ software instead of R. Hence, to design a cross-validation procedure was infeasible for this research. Besides, when the hyperparameters are selected by trial-and-error, they may not be optimal but we are still able to assess the behaviour of IHT.

The data is split into a train and test set (75%/25%). The algorithm is evaluated based on the predictive performance in the holdout set, i.e., the test set. The hyperparameters are selected based on performance measures in the training set, without making a distinction between a training set and validation set within the first training set, as this was also infeasible within the given time frame. Firstly, let us consider the hyperparameter for the lasso models in Table C.1 and Table C.2. Lasso requires one hyperparameter, namely the variance for the prior distribution, σ_β^2. For the LR model, remark that the AUC is highest for $\sigma_\beta^2 = 1$, while $\sigma_\beta^2 = 0.01$ is selected. The AUC measures in the test set of these models are, in order of smallest variance to largest, 85.31, 86.14, 86.33, 85.05. The last model with $\sigma_\beta^2 = 10$ did not converge. Hence, the variances $\sigma_\beta^2 = 0.1$ and $\sigma_\beta^2 = 1$ lead to an overfit of the model in training. For instance, the AUC for $\sigma_\beta^2 = 1$ is 89.75 in traning, and dropped significantly to 85.05 in testing. While we do not want to select the hyperparameters based on the test set, we would be able to avoid overfitting, if it had been feasible to construct a validation set. Hence, we choose the variance that produces the highest AUC in training, given that it does not cause overfitting. As a result, $\sigma_\beta^2 = 0.01$ is selected. Lastly, for Cox there was no

overfitting, hence the selection is based on the C-Statistic in training (see Table C.2).

Table C.1. Lasso for Logistic Regression

σ_β^2	AUC
0.001	85.03
0.01	**86.51**
0.1	88.14
1	89.75
10	-

Table C.2. Lasso for Cox Regression

σ_β^2	C-Statistic
0.001	85.24
0.01	**85.44**
0.1	84.83
1	84.20
10	84.25

Subsequently, we consider the hyperparameters for the IHT algorithm in the LR model. First of all, we have a hyperparameter for the initialising step of the algorithm, i.e., the initialising variance σ_β^2 for the Laplace distribution. The value of this hyperparameter has no effect on the selection of covariates nor the estimation of the coefficients, and hence, does not change the AUC in training. Therefore, we choose the same variance as was selected for the lasso model: $\sigma_\beta^2 = 0.01$. The same holds for the step-halving extension; it does not alter the estimation of the coefficients. Hence, the selection of hyperparameters for IHT with step-halving is identical for IHT without step-halving, regardless of the inclusion of the screening extension. In other words, the selection for IHT_1 is identical to the one for IHT_0, and the selection for IHT_3 is identical to the one for IHT_2.

Then, the only algorithms that require a hyperparameter in addition to the initialising variance are IHT_2 and IHT_3. We present the selection for IHT_2, but note that the selection is identical for IHT_3. For each value of k, we run the algorithm with four different values for the screening variance ς^2 and select the candidate with the highest AUC in training (see Table C.3).

Table C.3. IHT_2 for Logistic Regression

k	ς^2	AUC
5	0.0001	80.61
	0.001	**81.10**
	0.01	80.45
	0.1	80.35
10	**0.0001**	**82.16**
	0.001	81.53
	0.01	81.69
	0.1	80.60
20	0.0001	82.97
	0.001	**83.26**
	0.01	82.03
	0.1	81.73

Lastly, we address the hyperparameter selection for the IHT algorithm in a Cox setting. Similar to the LR model, the step-halving extension has no influence on the covariate selection or coefficient estimation. Hence, we only report the hyperparameter selection for IHT_0 and IHT_2; they are included Table C.4 and Table C.5, respectively. For IHT_0, the initialising variance σ_β^2 is the same for all values of k. The same holds for the screening variance ς^2 and the initialising variance σ_β^2 of the IHT_2 model.

Table C.4. IHT_0 for Cox Regression

k	σ_β^2	C-Statistic
5	**0.001**	**72.19**
	0.1	70.39
10	**0.001**	**71.94**
	0.1	69.74
20	**0.001**	**80.10**
	0.1	73.10

Table C.5. IHT_2 for Cox Regression

ς^2	σ_β^2	C-Statistic		
		$k = 5$	$k = 10$	$k = 20$
0.0001	**0.001**	**76.32**	**79.94**	**80.57**
0.0001	0.1	74.34	77.60	**80.57**
0.001	0.001	69.57	70.97	79.50
0.001	0.1	71.80	74.02	75.83
0.01	0.001	69.57	72.68	79.50
0.01	0.1	70.39	70.65	72.56
0.1	0.001	69.55	70.84	80.05
0.1	0.1	69.57	70.25	73.20

Appendix D Covariate Selection and Coefficient Estimates

Table D.1. Covariate selection and coefficient estimation for the IHT model with $k = 5$.

Domain	Covariate		LR		Cox	
			IHT_0	IHT_2	IHT_0	IHT_2
Demographic	Age		9.511	9.624	9.245	8.416
Conditions	Malignant neoplastic disease	(short term)	1.206	0.899	0.774	0.679
	Malignant neoplastic disease	(long term)				0.884
	Neoplasm of respiratory tract	(long term)		1.773		
	Primary malignant neoplasm	(long term)			1.112	
Drugs	Antiemectis and antinauseants	(long term)	1.923			
	High-ceiling diuretics	(long term)				0.699
Measurements	Oxygen saturation in arterial blood	(short term)			0.768	
Other	Charlson index - Romano adaptation*		3.115	3.024	2.684	
	CHADS2**					0.912

* The Charlson index is a comorbidity index

** CHADS2 carries information about the risk of experiencing a stroke

Table D.2. Covariate selection and coefficient estimation for the IHT model with $k = 20$.

Domain	Covariate		LR		Cox	
			IHT_0	IHT_2	IHT_0	IHT_2
Demographic	Age		9.672	9.199	8.217	8.001
	Female		-0.279	-0.269	-0.300	-0.340
Conditions	Malignant neoplastic disease	(short term)	0.862	0.955	0.620	0.699
	Malignant neoplastic disease	(long term)			0.259	0.723
	Neoplasm of respiratory tract	(long term)		1.714	1.166	
	Primary malignant neoplasm	(long term)	0.685			
	Primary malignant neoplasm of trunk	(long term)	0.826			
	Inflammatory disorder of head	(long term)			-0.239	
	Heart failure	(long term)				0.190
	Ulcer of skin or mucosa	(long term)		0.854	0.589	
	Traumatic injury	(long term)				0.170
	Abnormal renal function	(long term)		0.512		
Drugs	For alimentary tract and metabolism	(long term)			0.412	0.361
	Antiemetics and antinauseants	(long term)	1.570			
	Antiinflammatory and antirheumatic	(long term)			-0.260	-0.287
	Antipsychotics	(long term)	0.752	0.747	0.585	
	Beta blocking agents	(long term)		-0.192	-0.277	-0.249
	Apixaban	(long term)	-0.628			
	Bumetanide	(long term)		0.780		
	For obstructive airway diseases	(immediate)				0.248
	Endocrine therapy	(long term)	0.402			
	High-ceiling diuretics	(long term)			0.425	0.390
	Opioids	(long term)			0.439	0.412
	Prednisolone	(long term)				0.254
	Vitamin A and/or D	(long term)			0.292	0.312
	Vitamin K-1	(long term)		0.909		
	All other therapeutic products	(long term)	1.604			
Measurements	Body temperature	(long term)				0.281
	Oxygen saturation in arterial blood	(long term)				0.145
	Oxygen saturation in arterial blood	(short term)	0.737	0.718	0.592	0.365
	Cholesterol	(long term)			-0.154	-0.106
	Glomerular filtration rate	(long term)	-0.385	-0.463		
	Systolic blood pressure	(long term)			0.372	0.340
	Vitamin D and metabolites	(long term)	0.502	0.485		
Observations	Requests euthanasia	(long term)	0.632	0.692	0.573	
	Weight loss	(long term)	0.769			
Other	Charlson index - Romano adaptation*		2.690	2.532	1.744	
	CHADS2**					0.449

* The Charlson index is a comorbidity index
** CHADS2 carries information about the risk of experiencing a stroke