Statistical and Machine Learning Methods for Risk Prediction in Health

John Lenard Mbotwa

School of Medicine

University of Leeds

Submitted in accordance with the requirements for the degree of

Doctor of Philosophy

August, 2022

The candidate confirms that the work submitted is his own, except where work which has formed part of jointly-authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

Publications

Chapter 5 contains work based on the following publication: Mbotwa JL, Kamps Md, Baxter PD, Ellison GTH, Gilthorpe MS (2021) Latent class regression improves the predictive acuity and clinical utility of survival prognostication amongst chronic heart failure patients. PLoS ONE 16(5): e0243674. https://doi.org/10.1371/journal.pone.0243674

John L. Mbotwa analysed the data and drafted the manuscript. Other co-authors revised the manuscript.

This thesis is dedicated to my late Dad, Mr Lenard Mbotwa.

Thanks for teaching me to work hard.

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Professor Mark Gilthorpe, Professor Paul Baxter and Dr Marc de Kamps for their advice and support from the time I started my PhD up to now. Without their unwavering support, this work would not have been possible.

I would like to say thank you to Professor George Ellison who has been very supportive through his advice and encouragement.

Let me also thank the Commonwealth Scholarship Commission for funding my PhD. Without their financial support, I wouldn't have reached this far.

Thanks to my family, my mum, Florence Kaonga, my brother Owen, my sisters, Lumie, Mercy and Memory, my nephews, Calvin, Noel, Precious and Owen Jnr and lastly my niece, Mary for their love and support.

Lastly, I would also like to extend my gratitude to my wife Thumbiko and to my kids Claire and Raymond for the support and encouragement. You are my greatest source of motivation.

Abstract

Prediction of occurrence of an event in a patients' lifecourse is gradually becoming very important in this era of stratified medicine. With the availability of vast amounts of data in the form of Electronic Medical Records (EMRs), many risk prediction models (RPMs) have been developed for use in predicting future events in a patients' journey. RPMs use joint information collected from multiple predictors to provide a prospective insight into future 'potential' outcomes. Recent research developments indicate that there is a keen interest amongst researchers to develop RPMs that can be used to predict future events using routinely available information with optimum accuracy. Improvements in the prediction accuracy of RPMs would provide better quality guidance to health care policy makers in decision making process. Most of RPMs suffer from methodological shortcomings due to the inherent heterogeneity which causes patients to have different underlying risk profiles and therefore respond differently to treatment. Ignoring heterogeneity can affect the performance of RPMs which may lead to bias and poor estimation of the underlying risk for individuals. This thesis explores the benefits of using causal reasoning combined with latent variable methods to systematically improve prediction modelling. Throughout the thesis, the potential benefit of incorporating causal assumptions while predicting health outcomes is introduced through a lifecourse perspective using simulated datasets. Specifically, the thesis examines a latent class Cox proportional hazards (PH) model compared to the standard statistical modelling approaches typically adopted that do not explicitly accommodate population heterogeneity. The thesis also compares the Cox neural network approach which uses machine learning principles against the latent class Cox PH model. Lastly, this thesis explores the idea of predicting change, which is a composite outcome, using simulated datasets representing different possible data-generating scenarios and how this can enhance the RPMs.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Risk prediction modelling in Health

A risk prediction model (RPM) is a mathematical model that uses patient data (e.g. patients' demographic information, type of medication, genetic information etc.) obtained from a research study to estimate the probability of a patient experiencing a particular outcome (e.g. death or disease onset) in the future (Grant et al., 2018). Prediction of occurrence of events (e.g. mortality, hospital admission and readmission, disease onset, critical events in intensive care units (ICUs), etc.) is increasingly becoming important in medical research. With the availability of huge amounts of data in the form of Electronic Medical Records (EMRs), many risk prediction models have been developed for use in predicting future events in a patient's lifecourse. With the escalating costs related to the delivery of care, developing accurate prognostic RPMs would offer massive help by providing guidance to physicians and health care policy makers in decision making on treatment allocation, especially amongst patients in high risk groups. This would help in

allocation of resources to patients in need and subsequently reduce the overall cost of care.

There are different regression methods that are widely used to develop risk prediction models. The choice of method depends on the type of outcome under consideration. The outcome can be binary (e.g. death), categorical (e.g. blood group) or continuous (e.g. blood pressure or time to disease onset). The common regression models that are used for risk prediction in medical research are linear regression models, e.g. (Gaudart et al., 2004), logistic regression models e.g. (Zemek et al., 2016), and Cox proportional hazards regression models, e.g. (Sabouri et al., 2020). This is what constitutes the *traditional* modelling methods.

Traditional modelling methods assume homogeneity in the population (i.e. the relationship between an outcome and independent predictors is assumed to be the same for the whole population), which is not usually the case, especially when the population is heterogeneous.

This is only compounded when RPMs combine information from multiple predictors, to jointly provide prospective insight into future *potential* outcomes, e.g. 30-day mortality in patients suffering from an acute myocardial infarction (Steyerberg and Vergouwe, 2014), but each covariate might not behave exactly the same across all individuals of the population.

Multiple linear regression (MLR) analysis is a method used for examining the relationship between a single dependent variable (which in its basic form is assumed to be continuous) and a collection of independent variables (i.e. predictors) (Aiken et al., 2012). The independent variables may be quantitative (e.g. age, height, and weight) or categorical (e.g. sex) and do not need to follow any underlying distribution.

An extension to this is the generalised linear model (Skrondal and Rabe-Hesketh, 2004), which invokes a transformation of the outcome, by what is known as a link function, to accommodate other outcome distributions, the most common of which is encountered in health research is the logistic regression model for binary outcomes (Dreiseitl and Ohno-Machado, 2002). The Cox proportional hazards model is commonly used to assess the relationship between a time-to-event outcome (e.g. survival) and the independent model covariates (Ohno-Machado, 2001). This is achieved by relating the log of hazards function to the linear model with additive covariates. The Cox model is regarded as a semi-parametric model because it does not assume any distribution for it's baseline hazard function (i.e. a function that defines the instantaneous risk of the event, e.g. death)

The two main goals in risk prediction modelling are prognosis and diagnosis (Hendriksen et al., 2013). The ultimate goal in prognosis is to use the RPM to estimate the probability of a patient experiencing a clinical outcome (e.g. death); while in diagnosis, the RPM plays a role in identifying patients that are at risk of developing a particular condition. These two roles are very important in clinical practice as they are key to informing patients about their condition as well as in guiding potential therapeutic management needs.

1.2 Limitations of statistical methods that are commonly used for risk prediction in Health

Despite their importance in decision making processes, most RPMs suffer from methodological shortcomings. In the examples below, we discuss some of the limitations associated with RPMs.

1. Misspecification of covariate-outcome associations: This mostly arises due to inherent heterogeneity in the population of patients under study. Model misspecification may lead to the wrong inferences surrounding the estimated risk of patient outcomes. For example, in the presence of heterogeneity, patients respond differently to treatment. As such, using a standard model to estimate the risk estimates for the whole population may yield biased predictions. Heterogeneity in observational studies has given rise to individualised or personalised medicine, often termed 'Precision Medicine' (Currie and Delles, 2018). There are, however, epistemological limitations in how individualised prediction models will ever be achieved, even using more sophisticated approaches such as machine learning (Wilkinson et al., 2020). Addressing population heterogeneity thus remains a huge challenge in risk prediction modelling. Population heterogeneity may exist due to several unmeasured (i.e. unobserved) factors operating, which cannot be taken into consideration by the RPMs directly (i.e. these factors cannot be included as model covariates or interactions between covariates to adequately explain the inherent variation in the outcome). Ignoring the heterogeneity that exists in the population causes lack of precision in predictions sought and might lead to bias and underestimation of the underlying risk.

2. Sensitivity to missing data: A sufficient sample size is one of the key requirements in prediction research. RPMs require sufficient data to ensure that the predictions are accurate. The presence of missing information in the predictors may lead into inaccurate predictions and it may also affect the overall performance of RPMs. However, If missingness is at random the problem may be corrected by the method of multiple imputation to enable RPMs to yield unbiased estimates (Donders et al., 2006).

3. Choice of predictors: When the number of predictors available for selection is large, choosing the most parsimonious set of predictors to be included in the RPM, to avoid overfitting, becomes a difficult task to do. The choice of predictors to be included in a prediction model may be driven by expert opinion, through a review of past literature, or through the use of algorithms e.g. all possible subsets regression, stepwise methods like forward selection and backward elimination (Hocking, 1976). In many cases, the predictors are determined by first assessing their univariable associations with the outcome, but this is not ideal because it is the joint information of all selected predictors that is important, not the role any one isolated predictor (Arnold et al., 2020).

4. The inability to support robust causal inference in observational data: To generate robust causal inference from non-experimental (i.e. observational) data, there is need for a careful consideration of covariates acting as potential confounders and colliders to estimate the (potential causal) relationship between a specified exposure and a specified outcome (Tennant et al., 2017).

This is essential to ensure that neither the sampling nor the analyses introduce analytical or inferential bias through inappropriate conditioning on covariates (either deliberately or inadvertently). Even when covariates are accurately classified and appropriately treated in models designed to generate causal inference, unadjusted and residual confounding (from unmeasured or imprecisely measured confounders, respectively) make these models vulnerable to bias. Such considerations are largely irrelevant in multivariable prediction modelling, where the accuracy is derived from the joint information available from measured covariates (regardless of whether these are direct or indirect causes of the target variable and alternative covariate selection and parameterisation procedures are used to optimise the performance of prediction models). This means that multivariable prediction modelling does not need to pay attention to the analytical and inferential biases that can undermine causal inference models, except where counterfactual prediction is of interest (Sperrin and McMillan, 2020; Sperrin et al., 2018)

1.3 Possible ways of addressing heterogeneity in predictive modelling

The limited precision of standard methods in the presence of heterogeneity can be overcome by adopting statistical methods that try to identify hidden subgroups (i.e. clusters) of patients with similar attributes when modelling outcomes (Cochran et al., 2017). An example of a statistical procedure that relaxes the assumption of population homogeneity that is assumed under the traditional regression approach is the latent class regression (LCR) model. LCR models invoke clustering to identify unobserved heterogeneity in observational data (Magidson and Vermunt, 2004). LCR models can further be extended by integrating with causal knowledge in contexts where population heterogeneity is prominent, to embrace the cumulative consequences of variation which could also provide novel insights into subgroup differences that may substantively improve the accuracy of individual-level predictions.

Alternatively, machine learning (ML) methods are also gaining popularity in predictive modelling. ML can use artificial neural networks to find trends or relationships amongst variables, reduce dimensionality of 'big data', and to identify subgroups (i.e. clusters) based on the information extracted from the original dataset to facilitate prediction (Papachristou et al., 2016). These neural network algorithms provide an alternative to model-based LCR approaches to prediction and classification. Using neural networks, ML provide a potential alternative to latent class analysis with a few studies suggesting that they may offer significantly better predictive performance against most traditional approaches (Song et al., 2004; Zupan et al., 2000). So far, however there has been little investigation into:

- How LCR and ML methods might be compared (or combined in some way) for improved individual and subgroup prediction. Comparing ML methods against the LCR approach would help to identify ways of generating more reliable subgroups of patients with similar profiles and thereby improving prediction of the individual patient outcomes.

- Whether integrating causal reasoning in prediction modelling helps to build more reliable prediction models. Although this has been shown to work (Piccininni et al., 2020; Richens et al., 2020), this is largely under-research and it has not yet been considered whether prediction may be enhanced even further by embracing a lifecourse perspective.

1.4 Research aim

This thesis therefore explores different methods in clinical risk prediction context with interest in predicting change (in health status) as well as predicting discrete life events (e.g. death). Despite the former being rare in clinical prediction context, the interest of predicting *change* is not trivial. For this reason, we examine the challenges of predicting change when assessed through the lens of a causal framework and show, for the first time, how there may be more than one type of change outcome sought within a prediction framework. In either case, of predicting change or predicting time to discrete events, using causal knowledge to understand the underlying data generating processes may help to improve prediction and this thesis examines both scenarios.

This thesis further compares LCR modelling and ML methods in providing im-

proved prediction as well as to investigate whether incorporating causal reasoning in prediction modelling may help to build more reliable prediction models, especially if framed in a lifecourse perspective, where appropriate.

1.5 Research hypotheses

The research hypotheses are:

1. Using directed acyclic graphs (DAGs) to summarise causal associations amongst variables helps in the simulation process when addressing data from heterogenous populations by allowing the covariance structures for complex scenarios to be explored. Such simulation complexity may be overlooked if DAGs were not used in such instances.

2. Predicting change-score outcomes without including the baseline outcome as a predictor yields unreliable predictions; predicting the follow-up outcome is more robust. This is not generally appreciated and demonstrates the value of framing prediction from a causal thinking perspective.

3. Accounting for population heterogeneity by incorporating causal knowledge from a lifecourse perspective within a data generation process facilitates the identification of potentially clinically meaningful subgroups and helps to improve individual predictions.

4. Within a lifecourse causal framework, early-life covariates more strongly inform heterogeneity and subgroup classification in LCR models, whereas later-life covariates more strongly inform the outcome, and intermediate-life variable may inform both subgroup classification and the outcome.

5. LCR models may offer improved prediction over traditional regression or ML approaches when undertaking prediction on large and complex data that exhibits population heterogeneity.

6. ML's Cox neural network for survival modelling may offer improved predictions over standard regression or LCR Cox proportional hazards models when undertaking prediction on large and complex data.

1.6 Structure of the Thesis

Chapter 2 provides an overview of the statistical and machine learning methods applicable to this research. The emphasis is on the theoretical background of these methods and other technical considerations which form the backbone of the thesis.

Chapter 3 discusses how a directed acyclic graph (DAG) can be used to define the causal structure amongst variables to aid the data simulation process. The goal of this chapter is to establish ways of exploring complex data structures that are commonly encountered in observational studies in human health, through a carefully-considered simulation that reflects the underlying data generating mechanisms and not merely reflect its consequences as naively understood through subsequent data covariance structure. Two illustrations are considered in these simulations, the first illustration being an evaluation of prediction of *change* where the outcome is generated from subtracting a baseline measure from a follow-up measure (e.g. weight loss). The second illustration is the scenario of adopting a lifecourse approach for improved prediction of a time-to-event outcome (e.g. death).

Chapter 4 examines the first illustration of simulations carried out in Chapter 3 . The goal of this chapter is to examine the relationship between a baseline exposure and the subsequent change in a health outcome in an observational research setting. The objectives of this chapter are: to evaluate the impact of forcibly *including* the baseline, Y_0 as one of the predictors of either the change score, ΔY , or follow-up, Y_1; to evaluate the impact of forcibly *excluding* the baseline, Y_0 as one of the predictors of either the change score, ΔY, or follow-up, Y_1; to assess the implications of allowing the prediction model algorithm to select predictors from candidates Y_0 (i.e. baseline outcome), X_0 (i.e. baseline exposure) and U_0 (i.e. baseline competing exposure) while predicting either the change-score, ΔY, or follow-up, Y_1; and to assess the differences in root mean square error of the second objective within test data between both the change-score model and the follow-up model. A simulation approach is adopted for an illustrative example to aid understanding and to facilitate the evaluation of different plausible scenarios that may be encountered in real observational studies.

Chapter 5 explores four different statistical models to assess the predictive acuity and clinical utility when predicting survival amongst patients with chronic heart failure. The main goal in this chapter is to establish whether latent class regression models might outperform other standard modelling strategies in terms of accuracy and clinical interpretations in predictive modelling. We illustrate these issues using the UK-HEART study dataset.

Chapter 6 examines the second illustration of simulations carried out in Chapter 3. The goal of this chapter is to extend the latent class regression modelling by introducing the lifecourse concept. This chapter specifically aims at establishing the roles of different exposures throughout a patient journey and assesses

whether these may help to improve the prediction when using latent class regression models.

Chapter 7 examines the cox neural network for survival prediction. The aim of this chapter is to compare the Cox-nnet and LCR modelling for survival prediction. Different Cox-nnet architectures are explored to find an optimum architecture followed by an assessment of predictive acuity in each case. The generated results are compared against the standard Cox model and the LCR approach, specifically discussing how the integration of the LCR and neural network approaches might even further enhance predictive capabilities.

Chapter 8 summarises the results from the thesis in relation to the objectives. This is followed by limitations and suggestions for further research.

1.7 Contributions to the literature

The contributions to the literature arising from this thesis are as follows:

1. We have introduced DAGs to aid in defining causal structures to facilitate the simulation process, especially when the goal is to improve prediction of health outcomes. To our knowledge, no other authors to date have provided a detailed outline of the simulation process aimed at evaluating a prediction problem together with examples of predicting change and survival, while adopting a natural 'lifecourse' approach to the longitudinal data generating mechanism. This Chapter will therefore guide researchers in exploring different data generating mechanisms for complex scenarios in medical research.

2. Analysis and prediction of change is common, yet rarely is it appreciated that the use of change-scores is problematic. This has only recently been made clear for the causal interpretation of change-scores in observational data (Tennant et al., 2021a). This thesis builds upon this new work by examining the role of change-scores in the context of prediction. We have examined two regression methods that are commonly used to assess the relationship between the exposure(s) and the change in the outcome measured at two time points. The role of these methods in prediction has not been previously evaluated. Our conclusions from this work will guide researchers when considering similar scenarios.

3. We examined and compared four different modelling procedures to assess whether LCR models may offer improved prediction and clinical utility over traditional regression methods using real-world data. We concluded that LCR modelling can improve the predictive acuity of GLMs and enhance the clinical utility of their predictions. These methods have previously been used in association studies where the focus is to study the relationships between variables and outcomes, while in this application of these methods we are interested in improving predictions at the population and individual levels by accommodating heterogeneity within the models. This will add to the literature on methods to address heterogeneity and thus offer improved predictions.

4. We compared the LCR models against the recently proposed Cox-nnet model for survival prediction. These comparisons will help researchers to understand the methods better in terms of how each method works, the similarities and differences, as well as how these two methods might be integrated.

Chapter 2

Overview of the Statistical and Machine Learning Methods

2.1 Introduction

In this chapter, an overview of the statistical methods used in this thesis is provided. We begin by introducing fundamental aspects of survival analysis. The purpose of this section is to highlight some challenges associated with analysing survival data and to discuss why traditional regression methods may not be appropriate for analysing survival data. We then introduce the Cox model by first discussing its basic properties and assumptions and how parameters are estimated, followed by a section describing its extension to a latent class regression framework. We then discuss the technical aspects of the data generation process in carefully designed simulations that correspond to plausible causal structures that are common in real life studies. Before discussing the simulations in detail in Chapter 3, we first provide a brief background on causality and structural

equation modelling (SEM) as well as the associated thinking which underpins how DAGs are used to generate causally structured datasets. We also discuss the theoretical background and other technical details of the methods that are used in machine learning survival prediction modelling, (e.g. the Cox neural network).

2.2 General notation and Terminology

This thesis uses the following notation and terminology.

All random variables are denoted by upper case italic letters while the corresponding observed values are denoted by the lower case letters. For example, let $Y_1, \ldots, Y_n$ be n variables. The realizations of each of these variables are presented by $y_1, \ldots, y_n$. Vectors are denoted by bold lower case letters while matrices are denoted by bold upper case letters e.g. $\mathbf{y} = [y_1, \ldots, y_n]$ and $\mathbf{Y} = [Y_1, \ldots, Y_n]$ respectively. All matrices are presented in squared brackets. Greek letters are used to denote model parameters, e.g . $\beta_1, \ldots, \beta_n$.

2.3 Introduction to Survival Analysis

Survival analysis refers to the process of analysing the time until an event occurs. By event, we refer to occurrences like death, disease incidence, or any related experience that may occur during the period of follow-up. In survival analysis, the outcome variable of interest, which is also called Survival time, is the time from the beginning of follow-up until an event of interest happens. Survival time is usually presented in days, weeks, months or years depending on the objectives of the study. For example, a study might be interested in assessing survival for a

period of 5 years. In this instance, it might be more feasible to present survival time in years rather than weeks.

In practice, survival time is not always known for all patients due to censoring. Participants are said to be censored when information about their time-to-event is unknown (Prinja et al., 2010). This presents a key analytical problem in data analysis because the information about survival is incomplete. Censoring may happen either because some patients may have withdrawn from the study or because some patients are lost to follow-up during the study period, such that the only information about them is the time they were last in the study. Censoring may also happen simply because the patients have not experienced the event of interest during the finite follow-up period allocated.

One problem with survival analysis is that the time-to-event outcome variable does not follow a normal distribution. Time until an event of interest is always positive and often skewed, and it is therefore unreasonable to assume a normal distribution for a survival outcome. As such, the traditional ordinary least square estimation method cannot be used for analysis of survival. There are several methods in literature that are commonly used to analyse survival data. These include the non-parametric Kaplan Meier and the well-known Cox proportional hazards (PH) model. The Cox PH model is a semi-parametric model that makes fewer distributional assumptions about the hazard function compared to parametric methods. In parametric models, the functional form is completely specified, e.g. the Weibull hazard model.

The survival function is defined as the probability that a patient will survive

longer than some specified time t. It is represented mathematically as follows:

$$S(t) = P(T > t) \tag{2.1}$$

The hazard function, $h(t)$, is defined as the instantaneous likelihood at which events occur. It is represented mathematically as follows:

$$h(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T < t + \Delta t | T \geq t)}{\Delta t} \tag{2.2}$$

This numerator can be read as the conditional probability that the event will occur in the interval $[t, t + \Delta t]$ given that it has not happened. The denominator is an expression for the width of the interval.

2.3.1 The Cox PH Model

Let $\boldsymbol{X}^T = [X_1, X_2, \ldots, X_p]$ be a vector of p covariates or predictors for individual patient, $\boldsymbol{t} = [t_1, t_2, \ldots, t_n]$ be the survival times for the n patients. We can assess the relation between the distribution of survival time and $\boldsymbol{X}$ through a Cox PH model defined as

$$h(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{\beta}) = h_0(\boldsymbol{t}) \exp(\boldsymbol{\beta}^T \boldsymbol{X}), \tag{2.3}$$

where $\boldsymbol{\beta}^T = [\beta_1, \beta_2, \ldots, \beta_p]$ is a vector of parameters and $h_0(\boldsymbol{t})$ is an unknown baseline hazard function. The baseline hazard reflects the underlying hazard when the effect of the covariates is equal to zero. In other words when $X_1, X_2, \ldots, X_p$ is equal to zero.

2.3.2 Model assumptions

A key assumption about the Cox PH model is the proportional hazards function assumption (Hess, 1995). By assuming proportionality of the hazard function it means that each covariate or predictor has a constant multiplicative effect in the hazards function. In other words, the ratio of the hazards for two individuals from the same population remains constant over time.

2.3.3 Parameter Estimation

Cox proposed a partial likelihood approach for estimating the model parameters without necessarily specifying the distribution of the baseline hazard function. Suppose we have data with $(t_i, \sigma_i, \boldsymbol{X}_i)$ for individual i where t_i is the survival time, σ_i is the censoring indicator, $\boldsymbol{X}_i$ is the vector of covariates.

Let $\mathbb{R}(t_i)$ be the set of individuals who have neither experienced the event, nor been censored at $t = t_i$. In other words, these are individuals who are at risk of failure. Essentially, all individuals belong to the risk set $\mathbb{R}$ at $t = 0$ because all of them have not been censored at that point hence all of them are at risk of experiencing the event. The number of individuals in the risk set reduces when some individuals start experiencing the event of interest e.g. death. In other words, the number of individuals in the risk set reduces with time.

Suppose that the survival times are distinct with no ties such that $t_{(1)} < t_{(2)} < , \ldots, < t_{(n)}$, then the partial-likelihood function can be defined as

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} \frac{\exp(\boldsymbol{\beta}^T \boldsymbol{X}_{(i)})}{\sum_{j \in \mathbb{R}_i} \exp(\boldsymbol{\beta}^T \boldsymbol{X}_j)} \tag{2.4}$$

where $\boldsymbol{\beta}$ denotes the collection of unknown parameters to be estimated. These parameters can be estimated using the partial-likelihood approach (Cox, 1975). The partial-likelihood function is derived by taking the product of the conditional risk at time t_i given the set of individuals who have not failed or been censored by that time. A more detailed description for the partial-likelihood is given in the appendix section A.1.

2.4 Introduction to Latent Class Analysis and Latent Class regression Analysis

Latent class analysis (LCA) can be described as a statistical method used to classify individuals into unknown subgroups/clusters using measured or observed categorical or/and continuous variables. Individuals in the same subgroup tend to be similar with respect to the response patterns while those from different subgroups are different. This is also termed as traditional LCA.

Latent class regression (LCR) is an extension to the traditional LCA. Covariates are used to predict subgroup/latent class memberships. Each subgroup contains its own sub-model so that the estimated model parameters may be specific to that class (Gilthorpe et al., 2011). This is the main difference between LCA and LCR.

In LCA, the purpose is to assign individuals into latent classes. Individuals are assigned to latent classes according to their posterior membership probability which is determined using Bayes rule and this is called the *probabilistic assignment*. In-

dividuals can also be assigned to the latent class with the highest probability. This is termed *modal assignment*.

The optimum number of latent classes to be formed is typically determined by examining the Bayesian information criteria (BIC) as this has been shown to outperform other model fit statistics under simulations (Nylund et al., 2007). The ideal strategy for determining the number of classes may also be driven by interpretability decisions such as having clinical utility (Gilthorpe et al., 2011; Harrison et al., 2013; Kubzansky et al., 2014) and therefore needs to be considered and developed depending upon the context and application. The probability of any individual belonging to a particular class is based on the similarities in characteristics of individuals attributed to each class. Individuals may be probabilistically assigned to more than one class, with their total assignment over all classes summing to one.

One challenge with the LCR modelling approach lies in its estimation, which has to be achieved numerically and can be very sensitive to initial assumptions (i.e. starting values) used to maximize the likelihood function when estimating model parameters. If the starting values are far from optimum, the likelihood function fails to converge or takes longer to do so. For example, if 30 random starts are used, sometimes only 15 of them may give meaningful solutions when the likelihood function is maximized. For a solution to be meaningful, we expect the highest likelihood value to be replicated many times. When this fails, it means that the solution has not been achieved and one needs to increase the random starts to converge on a global optimum solution. The values that gave an optimum likelihood can be used as initial values for the final model in-order to reduce the search process (Muthén and Muthén, 2012).

2.4.1 The Latent Class Regression model

The latent class regression (LCR) model comes from a family of finite mixture models which classifies observations into classes to model unobserved heterogeneity within a population. Suppose that a population P is naturally partitioned into g classes $p_1, p_2, \ldots, p_g$. Let $\boldsymbol{y}$= $[y_1, y_2, \ldots, y_n]$ be an outcome variable from g distinct classes. Let the probability density functions for each of the g classes be $f_1, f_2, \ldots, f_g$ with corresponding proportions $\pi_1, \pi_2, \ldots, \pi_g$ for belonging to any of the respective classes. Thus the mixture density function of $\mathbf{y}$ is defined as

$$f(\boldsymbol{y}|\boldsymbol{Z}, \boldsymbol{X}, \lambda) = \sum_{i=1}^{g} \pi_i(\boldsymbol{Z}|\gamma_i, \boldsymbol{\delta}_i) f_i(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\beta}_i) \tag{2.5}$$

where $\lambda = (\gamma_i, \boldsymbol{\delta_i}, \boldsymbol{\beta_i})$ represents a collection of model parameters and $\pi_i(\boldsymbol{Z}|\gamma_i, \boldsymbol{\delta_i})$'s are class-membership probabilities that are estimated for each class and are dependent on a vector of covariates $\boldsymbol{Z}$ such that

$$\sum_{i=1}^{g} \pi_i(\boldsymbol{Z}|\boldsymbol{\gamma}_i, \boldsymbol{\delta}_i) = 1 \tag{2.6}$$

with $0 \leq \pi_i(\boldsymbol{Z}|\gamma_i, \boldsymbol{\delta}_i) \leq 1$ and $f_i(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\beta}_i)$ is the conditional probability density function for the observed response in the ith class model and $\boldsymbol{X}$ is the covariate vector.

For a class membership model, the structural part of the model is given by

$$\text{logit}(\pi_i(\boldsymbol{Z}|\gamma_i, \boldsymbol{\delta_i})) = \gamma_i + \boldsymbol{\delta}_i^T \boldsymbol{Z} \tag{2.7}$$

hence

$$\pi_i(\boldsymbol{Z}|\gamma_i, \boldsymbol{\delta_i}) = \frac{\exp(\gamma_i + \boldsymbol{\delta_i}^T \boldsymbol{Z})}{\sum_{j=1}^{g} \exp(\gamma_j + \boldsymbol{\delta}_j^T \boldsymbol{Z})} \tag{2.8}$$

where $\boldsymbol{Z}^T$=$[Z_1, Z_2, \ldots, Z_p]$ is a covariate vector for the class-membership model and $\boldsymbol{\delta_i}^T$ is the transpose of the vector $\boldsymbol{\delta_i}$ for the multinomial logistic class-membership model. Suffice to say, covariate vectors $\boldsymbol{X}$ and $\boldsymbol{Z}$ do not necessarily have to be the same.

2.4.2 The latent class regression Cox proportional hazards Model

We apply survival regression analysis within a latent class framework to predict subgroups of patients with different prognosis based on the available covariates. This is in conjunction with the prediction of survival distributions for different subgroups of patients using patient covariates. The distribution of the survival time variable for each component can be parametric (i.e. a scenario with distributional assumptions about the survival times), semi-parametric (i.e. a scenario with relaxed distributional assumptions), or non-parametric (i.e. a scenario without distribution assumptions about the survival times). If we assume a parametric model for the response variable, the component's densities are assumed to be from the same family. Some common distribution functions that may be considered appropriate for survival times in a parametric case include the exponential, Gamma and Weibull (Lee and Go, 1997). In a semi-parametric case, the Cox proportional hazard model is an example.

If $\boldsymbol{t}$ is a non-negative random variable representing time-to-death or time-to-loss-

of-follow-up or simply time to the end of the study for all patients with CHF, and suppose that for each individual we have a covariate vector, denoted $\boldsymbol{X}$ that affects the survival of patients in each class, we can define our survival model within a Latent class framework as follows:

$$S(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\theta}) = \sum_{i=1}^{g} \pi_i(\boldsymbol{Z}|\gamma_i, \boldsymbol{\delta}_i) S_i(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{\beta_i}), \tag{2.9}$$

where $\boldsymbol{\theta} = (\gamma_i, \boldsymbol{\delta_i}, \boldsymbol{\beta}_i)$ is the collection of parameters and $\pi_i(\boldsymbol{Z}|\gamma_i, \boldsymbol{\delta}_i)$ satisfies the constraints in 2.5. The vectors $\boldsymbol{X}$ and $\boldsymbol{Z}$ may include patient characteristics and medications. These covariates do not necessarily need to be the same in each class. If the effects of the covariates on the hazards (i.e. the instantaneous risk of event) in each class is constant during the entire duration of the follow-up period, then the hazard function can be specified as:

$$h_i(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{\beta}_i) = h_{0i}(\boldsymbol{t}) \exp(\boldsymbol{\beta}_i^T \boldsymbol{X}), \tag{2.10}$$

where $h_{0i}(\boldsymbol{t})$ is the baseline hazard for class i and $\exp(\boldsymbol{\beta}_i^T \boldsymbol{X})$ is the relative risk associated with a vector of predictors $\boldsymbol{X}$. We can derive a survival function from equation 2.10 as follows:

$$S_i(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{\beta_i}) = \Big[S_{0i}(\boldsymbol{t})\Big]^{\exp(\boldsymbol{\beta}_i^T \boldsymbol{X})} \tag{2.11}$$

where

$$S_{0i}(\boldsymbol{t}) = \exp\Big\{ - \int_0^t h_i(\boldsymbol{u}|\boldsymbol{x}, \boldsymbol{\beta_i})\, \mathbf{d}\boldsymbol{u} \Big\} \tag{2.12}$$

is the baseline survival for class i at time t given a vector of predictors $\boldsymbol{X}$ in that class.

2.5 Structural Equation Models and Wright's rules

Structural Equation Modelling (SEM) refers to a statistical technique that allows one to assess causal hypotheses on a set of observed and latent variables (Ullman and Bentler, 2012). Latent variables are variables that are not directly measured or observed but rather inferred from other variables. For example, the risk of developing a cardiovascular disease cannot be directly measured, but other factors e.g. (smoking status and age) may be used to classify an individual into the *high risk* or *low risk* subgroups. Observed variables are variables that we measure or observe, e.g. blood pressure, age, and smoking status.

Similar to traditional models, e.g. multiple regression, SEMs are based on an assumption of a linear relationship between the dependent and independent variables. This is because the observed variables are fundamentally assumed to be drawn from a multivariate normal, though this can be extended to other variable distributions through transformations. Therefore, any relationship between the variables is assumed to be linear. The linearity assumption is made because, It is simpler to demonstrate the methods principles for multivariate normal variables with linear relationships than other distributions, though non-normal variables and nonlinear relationships can be accommodated through variable transformations. From the analysis of observational data, the assumption of linearity must be validated through inspection of residuals (Gefen et al., 2000). The beta coefficients that are assigned to the causal graphs quantify the linear relationships

between variables.

Work on Structural Equation Modelling first emerged from the biologist, Sewall Wright (Wright, 1918, 1921a, 1934) who developed a path model with structural coefficients estimated on the basis of the correlation of both observed and latent variables. Wright introduced the method of path coefficients to show how correlations between variables can be used to quantify functional relationships between variables using a system of linear equations. Through his work on animal behaviour, Wright showed links between the correlations between variables and model parameters, and then demonstrated how the system of linear equations could be generated and used to estimate direct, indirect, and total causal effects of one variable on another variable (Tarka, 2018).

2.5.1 Wright's Rules for calculating a total association between any two variables in a path diagram

Sewall Wright (Wright, 1918, 1921a, 1934) proposed a set of rules for examining a path diagram that can help in generating a system of equations to describe relationships amongst a set of variables. The correlation between any two variables in a path diagram can be expressed as a contribution from all paths (i.e direct and indirect). The numerical contribution of an indirect path is the product of the path coefficients for each constituent arrow along the route. For any compound path:

- Loops are not allowed. This simply means that one cannot pass through the same variable twice when following a particular route.

- No going forward and then backward. This rule means that once one goes forward on a particular route or path, one cannot go backward to the variable(s) along the same or alternative backward route.

- A maximum of one curved arrow is allowed for each path. This rule allows for correlation, i.e. causal flow that is not explicitly specified; but curved arrows are allowed only the once for each path..

An illustration of Wright's Rules: Application to factors affecting wet bulb depression

To illustrate our understanding of Wright's rules for path analysis, we will go through an already existing example before applying the rules in our present context. Wright investigated causal factors which determine wet bulb depression (B). The factors Wright considered were temperature T, absolute humidity H and wind velocity (W). He also introduced radiation (R) as another factor correlated with all causal factors. Wind velocity was also assumed to be correlated with temperature and radiation as shown in Figure 2.1. Let $\beta_{BT} = t$ be the path coefficient which measures the relative influence of temperature on wet-bulb depression, $\beta_{BH} = h$ be the path coefficient which measures the relative influence of humidity on wet-bulb depression and $\beta_{BW} = w$ be the be the path coefficient which measures the relative influence of wind velocity on wet-bulb depression. Assuming that c, d, s, a, b are correlations between W and T, T and H, H and R, R and W respectively, we can use Wright's rules to find the following correlations between B and W, B and R, B and T, W and R W and T, R and T.

- The correlation between B and W: There is a direct effect from B to W

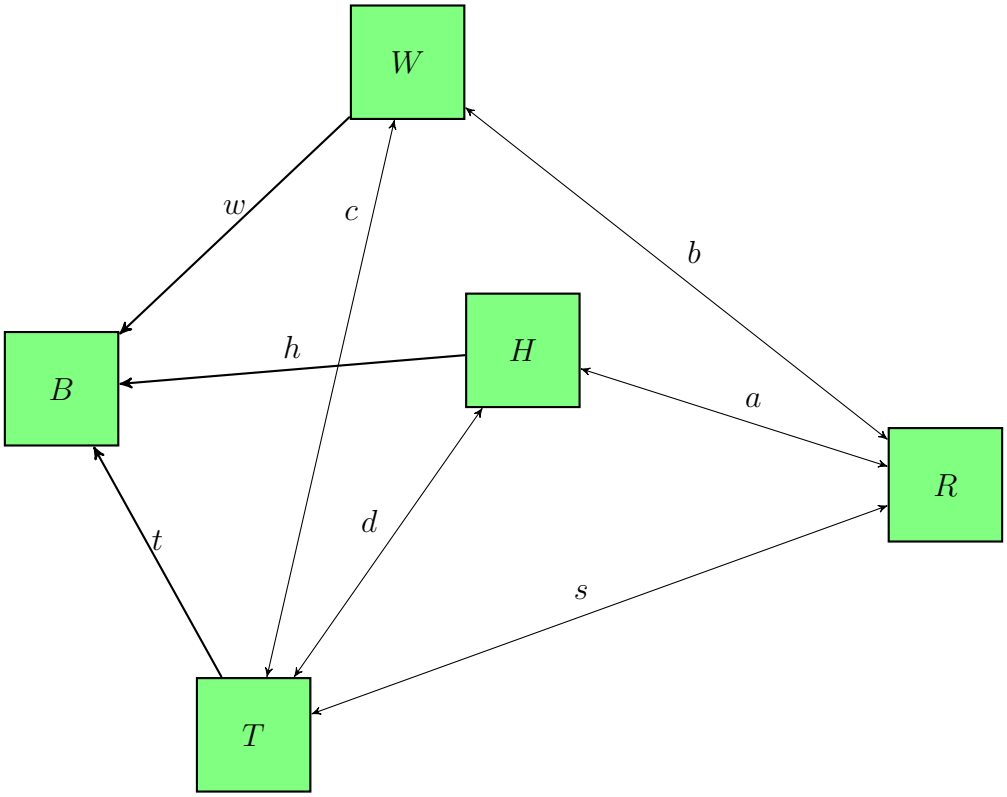


Figure 2.1: A path diagram depicting causal relations between wet-bulb depression (B), wind velocity (W), radiation (R), and temperature (T) taken from (Wright, 1921a).

represented by w. There is also an indirect path from B to W through T which can be represented by tc. Hence the total correlation between B and W is given by $w + tc$

- The correlation between B and R: There is no direct effect from B to R. There are three of indirect paths from B to R . The first path starts from B to R through T which can be represented as ts. The second path starts from B to R through W which can be represented as bw. The third path starts from B to R through H which can be represented as ah. Hence the total correlation between B and R is given by $ts + bw + ah$.

- The correlation between B and T: There is a direct effect from B to T represented by t. There are two indirect paths between B to T. The first indirect path starts from B to T through H which can be represented by dh. The second indirect starts path from B to T through W. Hence the total correlation between B and T is given by $t + dh + wc$.

- The correlation between W and R: There is a direct effect from W to R represented by b and there is no indirect route from W to R.

- The correlation between W and T: There is a direct effect from W to T represented by c and there is no indirect route from W to T.

- The correlation between R and T: There is a direct effect from R to T represented by s and there is no indirect route from R to T.

2.6 Structural Causal Models

The ideas relating to structural causal models (SCMs) were proposed by Pearl in the mid 90's (Pearl, 1995). In his work, he introduced nonparametric causal diagrams that may be used for identifying causal effects from non-experimental data.

In standard statistical analysis, one can use parameters estimated from regression models to infer associations amongst variables, estimate the likelihood of both past and future events happening as long as the experimental conditions remain static. However, in causal analysis the likelihood of events is examined under both static and variable conditions. This is the main distinction between a causal analysis and an associational analysis.

SCMs are defined as mathematical models that are used to represent causal relationships between variables, represented in the form of graphs called *directed acyclic graphs*. The Directed Acyclic Graph (DAG) can be viewed as causal path diagram that is used to study the relationships among a set of variables (i.e. exposure, outcome, confounders, and mediators). DAGs are called acyclic because they do not contain any loops. DAG's are the most recent nonparametric version of Wright's causal diagrams and parameterising a DAG under the assumptions of multivariate normality and linearity implies that Wright's rules then apply in DAGs.

Causal graphs forms an integral part of path analysis and structural equation modelling because they are used to summarise an investigator's assumptions about causal relations among variables.

Causal graphs are connected by unidirectional arrows (no bidirectional arrows like in standard path diagrams). Variables in causal graphs are called nodes and the connectors between variables are called edges or arcs.

The terminology used in causal graphs is similar to the one that is used in ordinary path diagrams. For example, given that there is an arrow from X_1 to X_2 ($X_1 \longrightarrow X_2$), where X_1 is a variable measured at time 1 and X_2 is another variable measured at time 2, then we would say that a variable X_1 directly affects a variable X_2. Similarly if three variables were related as follows ($X_1 \longrightarrow X_2 \longrightarrow X_3$), where X_2 is an intermediate variable measured at time 2 because it lies in the causal pathway between X_1 and X_3, we would say that X_1 indirectly affects X_3. The sequence of arrows from X_1 to X_3 is called a directed path or causal path. Any variable along a causal path from X_1 to X_3 is called a *mediator* variable. A variable X_1 may affect X_3 both directly and indirectly. For example, In this

diagram, X_1 affects X_2 directly and X_3 indirectly and that X_2 is a *mediator* of X_1 and X_3. The absence of a directed path between two variables represents the assumption that there is no causal link between them.

An example of a DAG

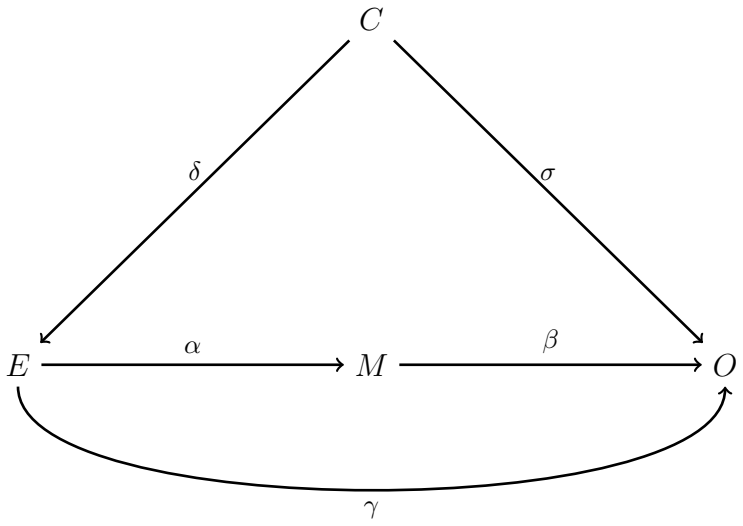


Figure 2.2: A DAG showing a confounder (C), exposure (E) , mediator (M) and an outcome (O)

2.7 Artificial neural networks

Artificial neural networks (ANNs) are constructs consisting of interconnections of nodes which act as information processing units synonymous with neurons in the human nervous system (Goodfellow et al., 2016). ANNs were first developed in 1943 to model non-linearities. The design of ANNs was motivated by the structure of a human brain which contain the neurons as information processing entities. There are two groups of ANNs that are common in medical literature. These are a) Feed forward neural networks (e.g. the multilayer perceptron) and Feed backward neural networks also known as recurrent networks (e.g. Hopfield networks). The main difference between these two groups is that the former does not have any loops while the latter include loops because of the feedback connections (Shahid et al., 2019). We will only focus on the multilayer perceptron in this chapter because the Cox neural network uses the principles of a feed forward neural network.

The basic rule in each of these ANNs is that a neuron in the network receives an input signal, processes it before sending out an output signal. Each node has at least one connection. Each connection has a weight coefficient that indicates the level of importance of the given connection in the neural network. A simplest architecture of an ANN comprises of three major layers of nodes, namely the input, the hidden layer and the output layer. An example of a detailed architecture of a single layer feed forward ANN is given in Figure 2.3. The input layer comprises the nodes formed from the list of the selected variables. The hidden layer contains hidden nodes which act as feature detectors for the neural network. A bias term is connected to every node in the hidden layer which can be interpreted as an

intercept in regression.

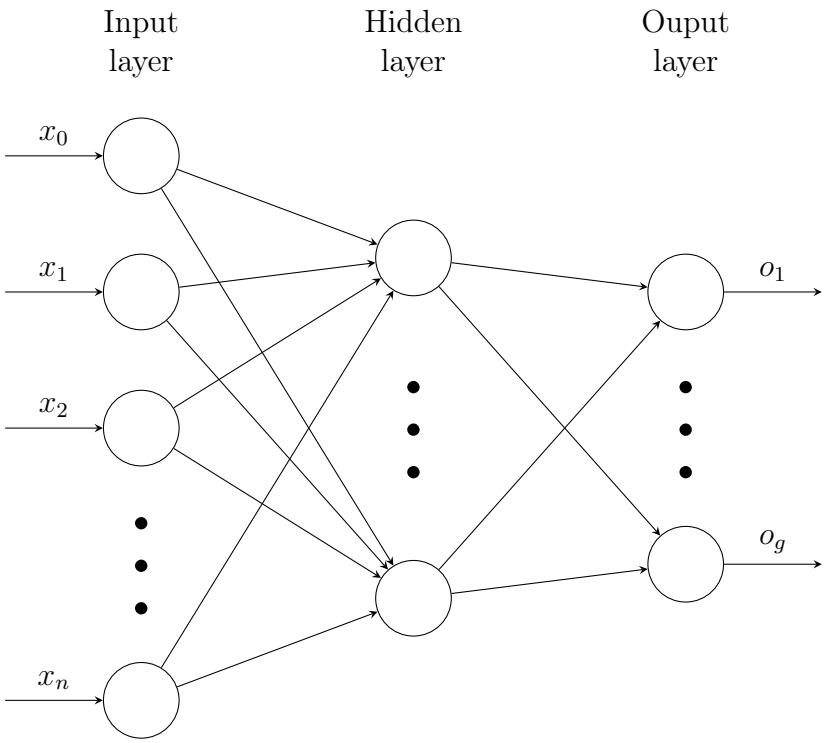


Figure 2.3: A typical feed forward neural network comprising three layers with n input nodes and k hidden nodes and g output nodes. x_0 is the bias term.

2.7.1 Training process

During the training process, the network tries to capture some unknown hidden information. There are two main types of training, namely supervised (i.e. the network processes the learning task by adjusting the weight coefficients until the desired result is achieved) and unsupervised also referred to as learning without a teacher (i.e. the network is not provided with the required output. The network explores the structure of the data, looks at any underlying correlations and uses these to organise patterns within the data).

Suppose that $X = \{x_1, x_2, \ldots, x_n\}$ is a set of input values, $H = \{h_1, h_2, \ldots, h_k\}$ is a set of hidden nodes and $O = \{o_1, o_2, \ldots, o_g\}$ is the set of output nodes, the network in Figure 2.3 can be trained to estimate an output $\hat{y}_{ij}$ for every subject i and node j in the training sample by minimising the loss function.. Data is fed forward through the network from the input to hidden and from hidden to output layer but not vice versa. During the training process, the hidden layer extracts important features from the training data by transforming the original input into a new space from which important features can easily be separated. Prediction or classification takes place on the output layer. Weights, w_i are initialised to every connection between the nodes. We seek values of the weights that make the model fit the training data well so as to minimise the loss function. For binary classification, the preferred choice is cross-entropy error (Kline and Berardi, 2005). The generic approach to minimising the function is by gradient descent through back-propagation (Li et al., 2012). Back-propagation is the process in which the network is fine-tuned by adjusting its weights until the error rate is minimised. In summary, an input is propagated forward via the hidden node(s) to the output layer where an error signal is calculated. If the error signal is significant, the signal is propagated back and weights adjusted accordingly. The process repeats until the the network is fully trained.

2.7.2 The detailed steps in back propagation

1. Data are fed into input nodes and its output is multiplied by the first set of connection weights then passed to the hidden layer

2. In the hidden layer, the incoming signals are summed up and transformed to an output that is multiplied by the second connection weight matrix then passed to the output layer;

3. In the output layer, the incoming signals are summed and transformed to produce the network output.

4. The difference between the output value and the target is assessed through the loss function and the error is propagated backward through network.

5. The connection weights are adjusted according to the loss function.

6. The process is repeated until the loss function is minimised.

The input of every node in the hidden layer is equal to the sum of the product of the weights w_i and the input values x_i plus a bias term b_k. In mathematical terms, we may describe the input and output at node j as

$$v_j = \sum_{i=1}^{n} w_{ij} x_j + b_j \tag{2.13}$$

and

$$\phi\Big(\sum_{i=1}^{n} w_{ij} x_i + b_j \Big) \tag{2.14}$$

respectively.

An activation function $\phi(v)$ is the function that is used to transform an input signal at each node to generate the output signal that is used as an input for the next layer. There are different types of activation functions that are commonly used in neural networks. The common ones are *sigmoid* and *tanh* activation functions (Sharma, 2017). The performance of different activation functions has

been previously studied. A *tanh* activation function was found to have a wider application and higher accuracy compared to other activation functions (Karlik and Olgac, 2011). The *tanh* activation function defined as

$$\phi(v) = \frac{\exp(v) - \exp(-v)}{\exp(v) + \exp(-v)} \tag{2.15}$$

is used as an activation function for a survival outcome prediction.

Artificial Neural networks are typically developed to model non-linearities that exist in complex datasets. For a network to capture as much non-linear relationships as possible, hidden nodes are added between the input and output nodes as shown in Figure 2.3. There are no standard rules available to help in choosing the correct number of nodes in the hidden layer. The choice is largely dependent on the number of inputs and the sample size. It may also be guided by the available background knowledge or through experimentation.

Adding too many nodes in the hidden layer inflates the chances of over fitting the data. An over-fitted model performs excellently on the training data but fails to perform if tested on a new dataset. Over-fitting may be avoided by adding a regularisation parameter (i.e. weight decay parameter) to the network's loss function which penalises larger weights according to the equation:

$$E_2 = E_1 + \lambda \sum w_{ij}^2 \tag{2.16}$$

where E_1 is a loss function and $\lambda \geq 0$. Larger values of the weight decay parameter tend to penalize larger weights more than smaller weights.

2.8 The Cox-nnet model

Cox-nnet model is an artificial neural network modelling framework that is an extension to the Cox PH model. Cox-nnet may be used to predict patient prognosis using high dimensional datasets (Ching et al., 2018). Cox-nnet is trained to minimise the partial log-likelihood defined as follows:

$$PLL(\boldsymbol{\beta}, W) = \sum_{i=1}^{n} \sigma_i \left[\boldsymbol{\beta}^T \phi(\boldsymbol{W^T X_{(i)}} + \boldsymbol{b}) - \log \sum_{j \in \mathbb{R}_i} \exp(\boldsymbol{\beta}^T \phi(\boldsymbol{W^T X_j} + \boldsymbol{b}) \right] \tag{2.17}$$

where

- σ_i is the censoring indicator for patient i.
- $\boldsymbol{\beta}^T$ is a vector for the regression coefficients.
- $\boldsymbol{W}$ is the coefficient weight matrix between the input and the hidden layer.
- $\boldsymbol{b}$ is the bias term for each hidden node.
- $\boldsymbol{X}_{(i)}$ is the covariate vector for patient i.
- $\phi(.)$ is the tanh activation function as shown in equation 2.15 and it is applied element-wise on a vector.
- $\|.\|$ is the L^2 norm. The L^2 norm is calculated as the square root of the sum of the squared vector values, e.g. Let $\boldsymbol{X}$=(x_1, x_2, x_3), $\|\boldsymbol{X}\| = \sqrt{{x_1}^2 + {x_2}^2 + {x_3}^2}$

- $\mathbb{R}_i$ is the risk set. A risk set is defined as the set of individuals at risk of experiencing an event of interest at a particular timepoint.

- $\boldsymbol{X}_j$ is the covariate vector for the patients in the risk set.

More details about the risk set and the partial log-likelihood function are given in Section A.1. Apart from being optimised for survival prediction, a Cox-nnet model may also be used to reveal useful biological information by analysing features extracted from the hidden layer nodes.

Mathematically, a single layer Cox-nnet can be described as a function $f : \mathbb{R}^n \mapsto \mathbb{R}^1$, where n is the size of the input vector. $\mathbb{R}^n$ is the n dimensional input vector and $\mathbb{R}^1$ indicates the single output layer which is the Cox regression layer. The output of the Cox-nnet is the Cox regression layer where the linear predictor ($\boldsymbol{\beta}^T\boldsymbol{x}$) of the Cox model is replaced by the outputs of the hidden layer as follows:

$$\theta_i = \boldsymbol{\beta}^T\phi(\boldsymbol{W}^T\boldsymbol{x_i} + \boldsymbol{b}) \tag{2.18}$$

Therefore, the output is the risk score, θ_i for each patient which represents the log hazards. This risk score is used to calculate the c-index for the Cox-nnet model.

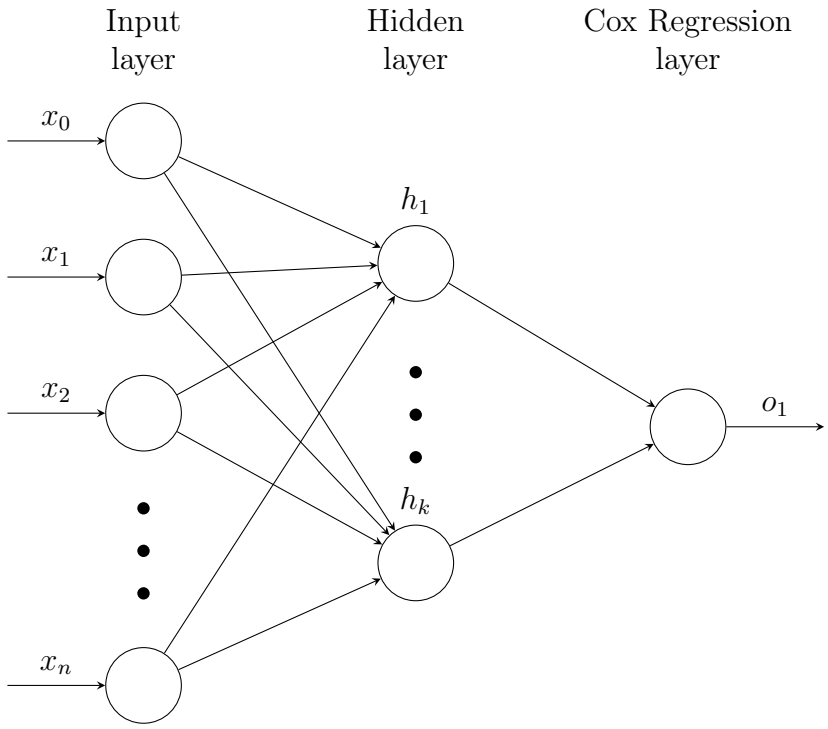


Figure 2.4: A general architecture of a single hidden layer Cox-nnet with n input nodes and k hidden nodes in the hidden layer and an output node also called the Cox regression layer. A bias term x_0 is connected to each node in the hidden layer.

2.8.1 The Cox-nnet software package

The Cox-nnet software package is used to implement artificial neural networks that is used to predict patient prognosis by extending Cox Regression to the non-linear neural network framework. It is built on the Theano math library. The main function for building a Cox-nnet survival model is called ***trainCoxMlp***. The ***trainCoxMlp*** function has six parameters which must be specified when building a Cox-nnet survival model. These are summarized below:

1. ***x_ train*** - A training set matrix where a Cox-nnet model is trained.

2. ***ytime_ train*** - Time to death or censoring for each patient in the training

set.

3. ***ystatus_train*** - Censoring status of each patient in the training set

4. ***model_params*** - Contains a dictionary of model parameters that are used when training a Cox-nnet model. It has the three parameters namely ***L2_reg*** which is used as a regularization parameter value, a ***node_map*** which defines the mapping of neurons in a network and as well as an ***input_split*** which shows how the input layer of the neural network is split.

5. ***search_params*** - Contains a dictionary of optimization hyper-parameters Below is an overview of all hyper-parameters associated with this function:

   (a) ***method*** : This is the algorithm used to perform gradient descent during parameter optimization process. Standard gradient descent (***gradient***) is used by a Cox neural network model to minimise a cost function (i.e. the partial log-likelihood function). The weights are adjusted after every loop until a local minima is attained.

   (b) ***learningrate***: The learning rate relates to how much the network weights should be adjusted in relation to the gradient of cost function. The value of the learning rate must be chosen carefully to avoid overshooting the steps. Smaller values of the learning rate are preferred, however this may also slow down the training process if the chosen value is too small. The default value in a Cox-nnet model is 0.01.

   (c) ***momentum*** : This is the proportion of momentum in momentum and nesterov gradients. The default value in a Cox-nnet model is 0.9.

(d) ***lrdecay*** : The decrease of the learning rate if the cost function is not decreasing. The default value in a Cox-nnet model is 0.9.

(e) ***lrgrowth*** : The increase of the learning rate if the cost function is decreasing. The default value in a Cox-nnet model is 1.0 (i.e. it does not increase. Adding a small term could, e.g. 1.01, could improve speed).

(f) ***evalstep*** - Number iterations between cost function evaluation in order to determine learning rate decay or growth. Setting this to a lower number will increase overhead. Default is 23.

(g) ***maxiter*** - The maximum number of iterations. The default value in a Cox-nnet model is 10000.

(h) ***stopthreshold*** - The threshold for stopping. If the cost does not decrease by this proportion, then allow the training to stop. The default value in a Cox-nnet model is 0.995.

(i) ***patience*** - The least number iterations a model should undergo before stopping. The default value in a Cox-nnet model is 2000.

(j) ***patienceincr*** - If a new lowest cost is found, wait at least *patienceincr* current iteration before stopping. The default value in a Cox-nnet model is 2.

(k) ***randseed*** - This is a random seed for initializing model parameters. The default value in a Cox-nnet model is 123.

6. ***verbose***- print more stuff if ***verbose*** =True.

2.8.2 Training the Cox-nnet models

The training process involves adjusting the weights to ensure that the predictions are optimized. The model parameters are updated in response to the output of the partial likelihood function. The loss function acts a guide to the optimization algorithm by providing feedback on how the training process is progressing.
One of the challenges with artificial neural networks in general is that a network may perform incredibly well in the training data but poorly in the testing data due to overfitting. This is a big problem because an overfitted ANN cannot perform well in a new dataset hence results cannot be generalized. One way to address this problem is through regularization (Srivastava et al., 2014). Regularization is a process by which extra information is added to the cost function to reduce overfitting and improve model performance. A penalty component is added to the loss function to penalize large model coefficients to correct overfitting.

2.9 Performance evaluation of the models

The concordance statistic (i.e. c-index or c-statistic) is commonly used to assess prediction performance in survival analysis (Harrell et al., 1982; Harrell Jr et al., 1984). A c-index measures the proportion of observations that are concordant, that is to say that the order of survival times and the model predictions are in agreement. The predictive information is derived from a set of predictors in the model. Patients with shorter survival times are supposed to have higher log hazard predictions while those with longer survival are supposed to have lower log hazards predictions. The c-statistic is similar to the area under the receiver operating curve, which is also a measure of discrimination for models with binary

outcomes. A c-statistic of 0.5 means that there is no predictive discrimination (i.e. the predictions are due to chance). A c-statistic of 1 means perfect prediction. Any c-statistic value above 0.8 is considered excellent.

The general procedure used to calculate the c-index advocated by Harrel et al (Harrell Jr et al., 1996) is given below:

2.9.1 General steps followed when calculating the c-index

The c-index is the proportion of all usable patient pairs in which the predicted and observed outcomes are concordant. Let $t_1, t_2, \ldots, t_n$ denote distinct survival times for the n patients and $r_1, r_2, \ldots, r_n$ denote the corresponding predicted risk (i.e. risk scores). c-index is calculated by considering pairs of patients in which atleast one one of them has experienced the event. The following are the conditions that must be satisfied

1. For each pair of patients (i, j) (with $i \neq j$), we look at their corresponding risk scores (r_i) and times-to-event (t_i). If both T_i and T_j are not censored, then we can observe when both patients experienced the event.

2. For each pair of patients (i, j) (with $i \neq j$), If the predicted risk score is smaller for the patient who lived longer (i.e. $r_i > r_j$ and $T_i < T_j$), the predictions for that pair are said to be concordant with the outcomes.

3. For each pair of patients (i, j) (with $i \neq j$), If the predicted risk score is larger for the patient who lived longer (i.e. $r_i > r_j$ and $T_i > T_j$), the predictions for that pair are said to be dis-concordant with the outcomes.

4. If both T_i and T_j are censored, then we do not know if they experienced the

event or not. If they did, we do not know who experienced the event first, and do not consider this pair in the computation of the c-index.

5. If the predicted risk is identical for a pair, $\frac{1}{2}$ rather than 1 is added to the count of concordant pairs in the numerator of c-index. Additionally, 1 is still added to the denominator of c-index.

6. If both patients experienced the event at the same time or if one patient experienced the event and the other hasn't been followed long enough to determine whether they will outlive the other, then the pair is considered unusable. We do not consider the pair in the computation.

7. Harrell's c-index is given by: c-index= $\frac{\text{concordant pairs}}{Total}$ where Total = concordant pairs + discordant pairs

2.9.2 Calculating the c-index for the Latent class Cox regression models

For a 2-class Cox proportional hazard model, the c-index was extended by exploiting the latent class structure through soft and hard clustering.
Hard clustering is a type of clustering where observations are allocated to a cluster or subgroup with the highest probability (i.e. modal allocation). Soft clustering on the other hand, is a type of clustering where observations are allocated to more than one subgroup (i.e. separate probabilistic allocations). The allocation in soft clustering is based on posterior probabilities. To calculate the c-index for the latent class Cox PH model, the following steps are used:

- Firstly, we ensure that the probabilistic and modal allocation is done.

- Generate the risk scores for each class.
- Generate the overall risk scores (R_1) in models where soft clustering was deployed. This was calculated as a weighted linear combination of the product of each score by its posterior probability for each class.
- Generate the overall risk score (R_2) in models where hard clustering was deployed. This was calculated as a weighted linear combination of the product of each score by its modal probability for each class.
- Use the general steps for calculating the c-index as outlined above to calculate the c-index using the soft clustering and hard clustering approaches.

Chapter 3

Using directed acyclic graphs (DAGs) to facilitate the data simulation process: An observation study

In this Chapter, we discuss how the data simulation process can be done by first using a directed acyclic graph (DAG) which was introduced in Chapter 2 to define the *causal* structure amongst the variables. The ultimate goal is to establish ways of improving the prediction process for a number of complex circumstances, though the principles of having a carefully considered simulation apply when evaluating any statistical process, not just improved prediction models. The two illustrations that form the basis of this chapter are evaluated in detail within the thesis in Chapter 4, when considering the prediction of *change* – the definition of which we contemplate carefully when considering, specifically the use of *change*

scores, and Chapter 6, where we consider improved prediction of a time-to-event outcome while adopting a lifecourse approach informed by causal reasoning.

3.1 Introduction

Simulation studies are useful when evaluating the performance, adequacy and properties of both current and novel statistical methods under a wide variety of settings (Bender et al., 2005; Crowther and Lambert, 2013). The validity of the simulations depends upon the way a data generation process has been specified. A poorly designed simulation study may lead to poorly simulated data which may then potentially affect the conclusions drawn from the statistical models being investigated. The existence of a suitable data generation process that accurately reflects the underlying causal relationships amongst variables improves the generalisation of the results from the simulations. A well-structured and carefully thought through data generation process can also guide researchers when interpreting results.

In survival analysis, simulations have been used to assess the performance of the parametric models (e.g. exponential, Weibull) as well as semi-parametric models (e.g. Cox proportional hazards model) when modelling time-to-event data (Bender et al., 2005). We seek to incorporate causal thinking in our data generation process so that our simulated data can reliably reflect true underlying hypothetical scenarios that give rise to the observed outcomes. We use causal path diagrams to simulate datasets that respect a causal processes. Where necessary and appropriate, we also consider latent structure to reflect more complex underlying features that might not be attributable to a single candidate predictor

phenomena, such as population heterogeneity.

3.1.1 Advantages of simulating using a DAG compared to simulating directly from models with the specific distributions and covariance structure

- The main distinction between simulations that respect the *causal* data generating process and the naive approach of merely reflecting context through the observed covariance structure is that the former closely emulates reality directly as opposed to the latter only emulating consequences of the underlying processes. Therefore, simulating from the latter does not reflect the mechanisms of data generation process, thereby potentially missing intrinsic latent but critical features that are only apparent if considered within a causal temporal perspective. Simulating from a DGM more narrowly specifies the underlying mechanism behind the observed variations that are intrinsic to the context.

- The DAG based approach incorporates the known causal structures of the assumed data generating mechanisms. This enables researchers to explore the range of plausible scenarios that reflect the reality of these data generating mechanisms. This is not guaranteed to be the case when simulating data from a specified covariance structure where the exploration of different parameter values need not map directly onto the postulated causal structure.

- To address heterogeneity in observational data, we introduced a lifecourse context to frame the Data Generating Mechanisms to contextualise the underlying processes that contribute to intrinsic population heterogeneity. This could not have been achieved by simulating data from the observed covariance structure.

- The interpretation of the results from DAG-based simulations is therefore more robust because it explicitly incorporates knowledge of the data generating structure, opposed to merely capturing the consequences of it, as would be seen cross-sectionally and upon which the simulations based on a covariance structure would achieve. The DAG based approach therefore has a greater scope to reflect the underlying reality.

3.1.2 Overview of the illustrative examples

1. The first illustration examines the deceptively challenging scenario of predicting the outcome described as *change*–specifically we examine the prediction of *change scores*, which are generated from subtracting a baseline measure from a followup measure. As this conflates the causal mechanisms of both measures into a single outcome, we must reflect upon the data-generating processes assumed and what is therefore ultimately the most robust concept of *change* to be predicted and any corresponding assessment of how good the prediction is.

Examining this context through a causal lens differentiates predictive acuity from mathematical artefact, thereby making clearer the objective and achievement of prediction sought for what we perceive to be the drivers of genuine *change*(not merely the misleading summary substitute offered by *change scores*).

2. The second illustration is more specifically seeking to improve prediction in the context of clinical risk prediction models (RPMs), where the illustrative example is that of survival among coronary heart disease patients within a heterogenous population. Focus is given to predictors that might be measured at different stages of the lifecourse to reflect the multiple impacts of different experiences throughout the lifecourse that compound to yield population heterogeneity in both the survival outcome and its relationship to candidate predictors.

   We therefore explore different data generating possibilities in terms of both population heterogeneity and the strength of predictor relationships to the ultimate outcome of death, described in a causal framework using a DAG. It is hoped that bringing a causal framework to bear on this context would both improve prediction capability and provide additional insight into the role of different predictors at various stages of the lifecourse.

3.2 Procedure taken when simulating data

1. **Theory**: In the first step, we hypothesize possible causal relationships between variables within an observational study setting.

2. **Developing a measurement model**: The second step involves translating the hypotheses generated in step 1 into a possible causal graph.

3. **Developing a structural model**: The third step uses the graph in part 2 to show how the variables are structurally related under the naïve assumptions of linearity and multivariate normality by assigning the *beta* coefficients to the causal graph to quantify the relationships between variables.

4. **Assess the covariance matrix**: Once a model has been specified, we assess the model to ensure that that the covariance matrix is semi-positive definite.

5. **Simulate data**: The fourth step is to simulate the data.

6. **Transform data where necessary**: Once data have been simulated, we transform variables (i.e. the outcome and/or predictor variables) to match the distributional properties of the data being sought (e.g. for a time-to-event outcome, we generate an exponential distribution for survival time and for a binary variable we simply dichotomise; other variables types may also be constructed as necessary by appropriate transformation).

7. **Check basic model statistics**: Lastly, we compute basic statistics to assess the simulated data relationships and contrast to real-world scenarios – where there is discrepancy, tweaks to the path coefficients in the causal graph may be pursued and steps 3 to 7 are repeated until a satisfactory causal graph, corresponding covariance matrix, and simulated data are achieved. If it is anticipated that any nonlinear relationships occur amongst variables, these must be incorporated in the model being evaluated in step

7 by making necessary transformations until the desired covariance matrix is obtained.

3.3 Prediction of Change: Simulation 1

Studies of changing phenomena are common in science, yet the methodological issues involved in the analysis of *change* for observational data are deceptively complex. For instance, the use and interpretation of *change-scores*, also called *difference scores*, *gain scores*, or *change-from-baseline variables*, is problematic due to change-scores being composite, i.e. constructed from two measures of a single parent variable (Y).

The composite change-score is determined by subtracting a future measure of the parent (Y_1, *follow-up*) from an earlier measure (Y_0, *baseline*), yielding a single assessment of *change*: $\Delta Y = Y_1 - Y_0$. Change-scores therefore contain tautological information about both determining parents and when examined in relation to other variables, it is not clear what is being evaluated. If researchers seek causal insight, for instance, *How do statins reduce the risk of heart failure?*, change-scores cannot yield meaningful insights within observational data (Tennant et al., 2021a).

A persistent confusion stems from the concept of what is inferred to be a useful assessment of *change*. As argued by Shahar and Shahar (2012), change-scores are not of causal interest (Shahar and Shahar, 2012); what matters is the *exogenous change* in the outcome, i.e. the part of Y_1 that cannot be explained by Y_0, which we can depict as C_Y. Thus, to enquire of a causal relationship between an exposure, X_0, and change in Y is to define the focus of interest to be an assessment of the impact of X_0 on C_Y, which is not the same as an assessment of the impact of X_0 on the change-score, $\Delta Y = Y_1 - Y_0$. The focus of interest is therefore esti-

mated by assessing the impact of X_0 on Y_1, controlling for Y_0 or not depending upon the causal relationship between Y_0 and X_0 (Tennant et al., 2021a).
If the analytics of *change* matter substantially for a longitudinal outcome when examined in relation to its putative causes, how, if at all, might it matter when we seek to predict what we describe as *change*? In a casual framework, it is clear that we should consider change to be assessed through the evaluation of the follow-up outcome, so how might this impact our view of how we evaluate and assess the merits of predicting *change*? Should we be concerned about the choice between predicting change-scores, or predicting follow-up outcome and calculating the change-score afterwards, if reporting summary statistics of the latter is important? To investigate this, mindful of the additional insights that a causal lens would bring to this prediction challenge, we simulated data for a few simplified scenarios to see what unfolded differently in relation to what we seek in the prediction of *change* according to both outcome options..

We thus examine a longitudinal study setting in which we are interested in predicting an outcome Y which is recorded at two time points, with values, Y_0 and Y_1 denoting the values of Y at baseline and at follow-up, respectively. We can calculate the change-score by subtracting the value recorded at baseline from that recorded at follow-up ($\Delta Y = Y_1 - Y_0$). Suppose that X_0 and U_0 are exposure variables measured at baseline. If the objective is purportedly to predict *change*, it might seem intuitive to seek to predict the change-score; we can do this using both X_0 and U_0 as candidate predictors. We have seen that it is also possible, and indeed better practice, to predict follow-up and calculate the change-score post-hoc (Senn, 2006). In our simulations, we can use DAGs to

define a range of scenarios that are plausible, as shown below. We define various DAGs with the four variables, X_0, Y_0, Y_1 and U_0. The path coefficients from $U_0 \longrightarrow X_0$, $U_0 \longrightarrow Y_0$, $U_0 \longrightarrow Y_1$, $X_0 \longrightarrow Y_0$, $X_0 \longrightarrow Y_1$ and $Y_0 \longrightarrow Y_1$ are represented by $\rho U_0 X_0$, $\rho U_0 Y_0$, $\rho U_0 Y_1$, $\rho X_0 Y_0$, $\rho X_0 Y_1$ and $\rho Y_0 Y_1$ respectively. We explore a range of plausible data generating processes with path coefficients defined as follows: $\rho Y_0 Y_1 \in \{0, \ldots, 0.95\}$; $\rho X_0 Y_1 \in \{-0.95, -0.90, \ldots, 0.95\}$; $\rho U_0 Y_1, \rho X_0 Y_0 \in \{-0.5, 0.5\}$; $\rho U_0 X_0$, $\rho U_0 Y_0 \in$ {-0.5,0,0.5}.

3.3.1 Dag 1: X_0-Y_0 orthogonal

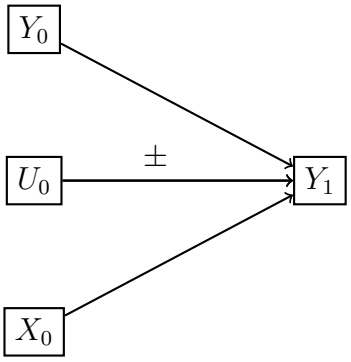


Figure 3.1: X_0-Y_0 orthogonal & no U_0 confounding.

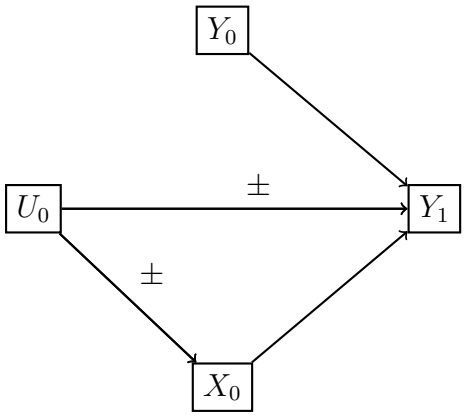


Figure 3.2: X_0-Y_0 orthogonal and U_0 confounds X_0.

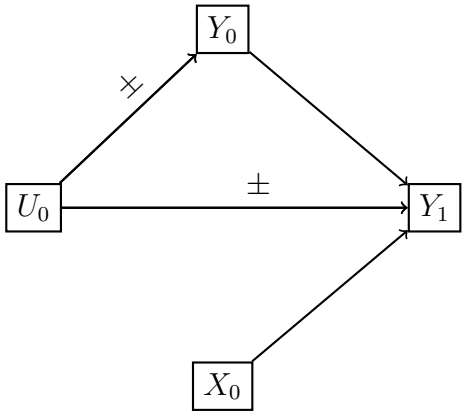


Figure 3.3: X_0-Y_0 orthogonal & U_0 confounds Y_0.

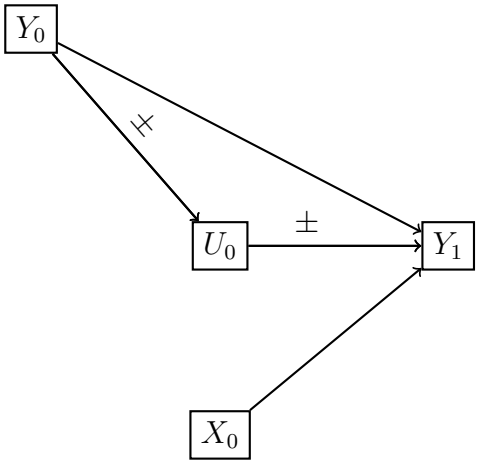


Figure 3.4: X_0-Y_0 orthogonal & U_0 confounds X_0 & Y_0.

3.3.2 Dag 2: X_0 confounds Y_0

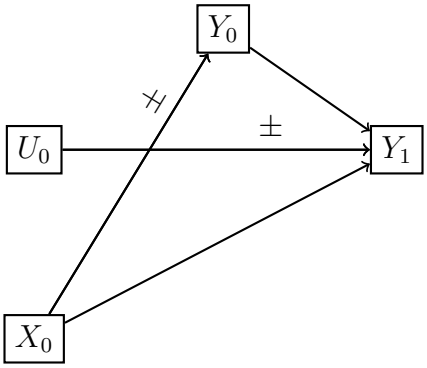


Figure 3.5: X_0 confounds Y_0 & no U_0 confounding.

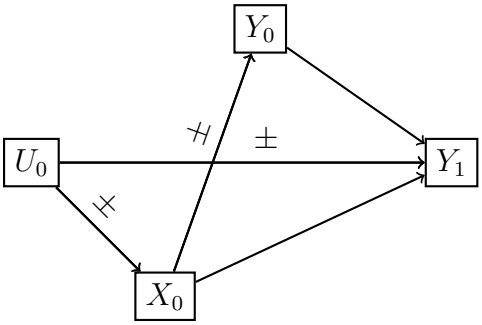


Figure 3.6: X_0 confounds Y_0 & U_0 confounds X_0.

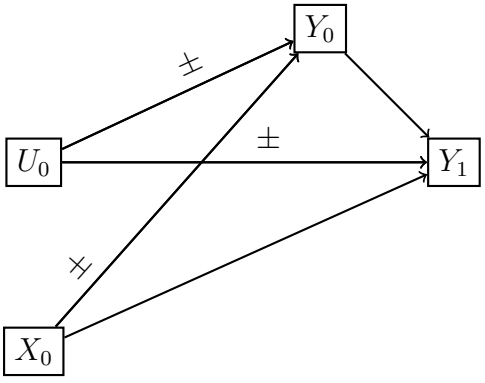


Figure 3.7: X_0 confounds Y_0 & U_0 confounds Y_0.

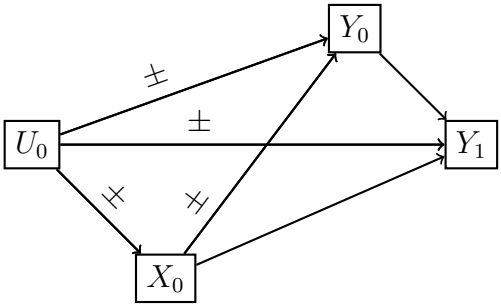


Figure 3.8: X_0 confounds Y_0 & U_0 confounds X_0 & Y_0.

3.3.3 Dag 3: X_0 mediates Y_0

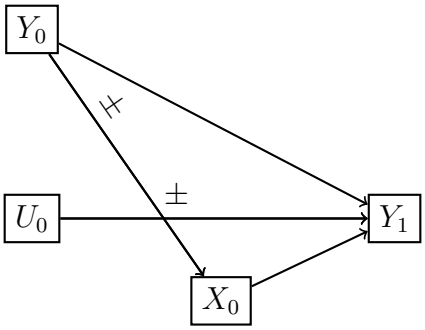


Figure 3.9: X_0 confounds Y_0 & no U_0 confounding.

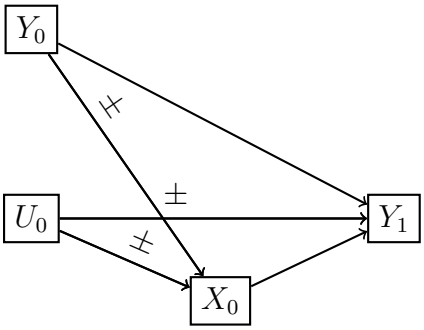


Figure 3.10: X_0 mediates Y_0 & no U_0 confounding.

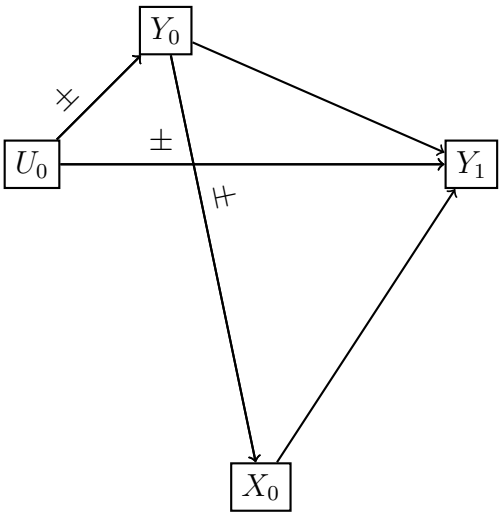


Figure 3.11: X_0 mediates Y_0 & U_0 confounds Y_0.

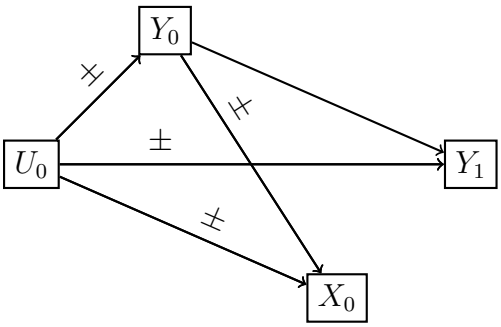


Figure 3.12: X_0 mediates Y_0 & U_0 confounds X_0 & Y_0.

3.3.4 Description of the DAGS

1. Dag 1: In Figures 3.1- 3.4, we assume that X_0 and Y_0 are orthogonal as there is no causal relationship between them, analogous to the situation for a randomised controlled trial (RCT) – although we are primarily interested in the context of prediction within observational data, RCT scenarios are important for completeness when making comparison across different potential underlying data generating structures. We generate four different scenarios as described below:

    - Scenario 1: In Figure 3.1, we assume that there is no confounding of X_0 and Y_1 by U_0. The correlation between U_0 and Y_1 is sampled from $\{-0.5, 0.5\}$. The correlations between U_0 and Y_0, U_0 and X_0 as well as X_0 and Y_0 are set to 0. The correlation between Y_0 and Y_1 is sampled from the set $\{0.05, 0.10, \ldots, 0.95\}$ and the correlation between X_0 and Y_1 is sampled from $\{-0.95, -0.90, \ldots, 0.95\}$.

    - Scenario 2: In Figure 3.2, we assume that U_0 confounds X_0 and Y_1, but does not confound Y_0 and Y_1. The correlation between X_0 and Y_0 and between U_0 and Y_0 is set to 0. The correlation between U_0 and Y_1 as well as between U_0 and X_0 is sampled from $\{-0.5, 0.5\}$. The correlation between Y_0 and Y_1 is sampled from $\{0.05, 0.10, \ldots, 0.95\}$ and that between X_0 and Y_1 is sampled from$\{-0.95, -0.90, \ldots, 0.95\}$.

    - Scenario 3: In Figure 3.3, we assume that U_0 confounds Y_0 and Y_1, but does not confound on X_0 and Y_1. We let the correlations between X_0 and Y_0 and that between U_0 and X_0 to be equal to 0. The correlations between U_0 and Y_1 as well as between U_0 and Y_0 is sampled from

$\{-0.5, 0.5\}$. The correlations between Y_0 and Y_1 are sampled from the set $\{0.05, 0.10, \ldots, 0.95\}$ and that between X_0 and Y_1 is sampled from $\{-0.95, -0.90, \ldots, 0.95\}$.

- Scenario 4: In Figure 3.4, we assume that X_0 and Y_0 are orthogonal and that U_0 mediates Y_0 and Y_1. The correlation between X_0 and Y_0 as well as between U_0 and X_0 to be equal to 0. The correlation between U_0 and Y_0 as well as between U_0 and Y_1 is sampled from $\{-0.5, 0.5\}$. The correlation between Y_0 and Y_1 is sampled from the set $\{0.05, 0.10, \ldots, 0.95\}$ and that between X_0 and Y_1 is sampled from $\{-0.95, -0.90, \ldots, 0.95\}$.

2. Dag 2: In Figures 3.5- 3.8, we assume that X_0 confounds Y_0

   - Scenario 1: In Figure 3.5, we assume there is no confounding of X_0 and Y_0 by U_0. We let the correlation between U_0 and Y_1 is sampled from $\{-0.5, 0.5\}$. The correlation between U_0 and Y_0 as well as between U_0 and X_0 are both set to 0. The correlation between Y_0 and Y_1 is sampled from the set $\{0.05, 0.10, \ldots, 0.95\}$ while the correlation between X_0 and Y_1 is sampled from $\{-0.95, -0.90, \ldots, 0.95\}$.
   - Scenario 2: In Figure 3.6, we assume that U_0 confounds X_0 and Y_1, but does not confound on Y_0 and Y_1. The correlation between U_0 and Y_0 is set to 0. The correlation between U_0 and Y_1 as well as between U_0 and X_0 is sampled from $\{-0.5, 0.5\}$. The correlation between Y_0 and Y_1 is sampled from $\{0.05, 0.10, \ldots, 0.95\}$ and that between X_0 and Y_1 is sampled from $\{-0.95, -0.90, \ldots, 0.95\}$..

- Scenario 3: In Figure 3.7, we assume that U_0 confounds Y_0 and Y_1, but does not confound X_0. The correlation between U_0 and X_0 is set to 0. The correlation between U_0 and Y_1 as well as between U_0 and Y_0 is sampled from $\{-0.5, 0.5\}$. The correlation between Y_0 and Y_1 is sampled from $\{0.05, 0.10, \ldots, 0.95\}$ and that between X_0 and Y_1 is sampled from $\{-0.95, -0.90, \ldots, 0.95\}$.

- Scenario 4: In Figure 3.8, we assume that U_0 confounds X_0, Y_0 and Y_1. The correlation between X_0 and Y_0 and that between U_0 and X_0 is sampled from $\{-0.5, 0.5\}$. The correlation between U_0 and Y_0 as well as between U_0 and Y_1 is set to ± 0.5. The correlation between Y_0 and Y_1 is sampled from $\{0.05, 0.10, \ldots, 0.95\}$ and that between X_0 and Y_1 is sampled from $\{-0.95, -0.90, \ldots, 0.95\}$.

3. Dag 3: In Figures 3.9- 3.12, X_0 mediates Y_0

   - Scenario 1: In Figure 3.9, we assume that there is no confounding of X_0 and Y_1 by U_0. The correlation between U_0 and Y_0 as well as between U_0 and X_0 is set to 0. The correlation between U_0 and Y_1 is sampled from $\{-0.5, 0.5\}$. The correlation between Y_0 and Y_1 is sampled from the set $\{0.05, 0.10, \ldots, 0.95\}$ while the correlation between X_0 and Y_1 is sampled from $\{-0.95, -0.90, \ldots, 0.95\}$.

   - Scenario 2: In Figure 3.10, we assume that U_0 confounds X_0 and Y_1, but does not confound Y_0 and Y_1. The correlation between U_0 and Y_0 is set to 0. The correlations between U_0 and Y_1 as well as between U_0 and X_0 are both sampled from $\{-0.5, 0.5\}$. The correlations between Y_0

and Y_1 are sampled from the set $\{0, 0.10, \ldots, 0.95\}$ and that between Y_0 and X_0 is sampled from $\{-0.95, -0.90, \ldots, 0.95\}$.

- Scenario 3: In Figure 3.11, we assume that U_0 confounds Y_0 and Y_1, but does not confound X_0 and Y_1. The correlation between X_0 and Y_0 and that between U_0 and X_0 is set to 0. The correlation between U_0 and Y_1 as well as between U_0 and Y_0 is sampled from $\{-0.5, 0.5\}$. The correlations between Y_0 and Y_1 is sampled from $\{0.05, 0.10, \ldots, 0.95\}$ and that between X_0 and Y_0 is sampled from $\{-0.95, -0.90, \ldots, 0.95\}$.
- Scenario 4: In Figure 3.12, we assume that U_0 confounds X_0 and Y_1 and U_0 confounds Y_0 and Y_1. The correlation between U_0 and Y_0 as well as between U_0 and X_0 is sampled from $\{-0.5, 0.5\}$. The correlation between U_0 and Y_0 as well as between U_0 and Y_1 is sampled from $\{-0.5, 0.5\}$. The correlation between Y_0 and Y_1 is sampled from the set $\{0.05, 0.10, \ldots, 0.95\}$ and that between X_0 and Y_0 is sampled from $\{-0.95, -0.90, \ldots, 0.95\}$.

3.4 Prediction of Survival or death in a heterogenous population: Simulation 2

We now consider a prediction challenge for longitudinal data, as might arise in many clinical prediction models. We therefore adopt a lifecourse perspective for our simulation study and seek to evaluate – within a hypothetical cohort of patients with a history of chronic heart failure – the role of exposures at different stages of the lifecourse in predicting both the outcome of death or survival, S, and unobserved population heterogeneity captured by a categorical latent class variable C. In this instance, we adopt a lifecourse framework to understand how different experiences encountered during different periods of an individual's life might predict the risk of death later in life and contribute to the accumulated variations that give rise to population heterogeneity. The initial step is to construct a DAG to represent our hypothesized causal effects among the observed variables and the latent variable for population heterogeneity; this will become the data generating processes for our simulations.

We consider three variables X_1, X_2 and X_3 representing exposures that have occurred at different times throughout an individual's life. We arbitrarily assume that X_1 represents all variables or attributes which pertain to a period during the early life of an individual. Examples of such variables are: genetics, birthweight, and socioeconomic background. Variable X_2 represents all variables or attributes of individuals which pertain to a period midway through an individual's life. Examples of these variables are: obesity, lack of physical exercise, and smoking. Finally we have another set of variables represented by X_3. These variables pertain to the period just before the outcome of interest. Examples could be body

size, comorbidities, treatment type, and drug adherence. In a DAG, variables X_1, X_2 and X_3 are represented by square boxes to indicate that these are measured variables. However, variable $\hat{C}$ is represented as a circle to indicate that this is not a directly observed variable but rather inferred from other observed or unobserved variables. In this instance, we might assume that variable $\hat{C}$ is binary with two categories that, post-hoc, we can describe as representing *high* and *low* risk of death within the population. Survival ($\hat{S}$) is the outcome variable that describes the period an individual is observed during the follow-up period till either death or the end of the study (i.e. censored).

A hypothetical scenario presented in Figure 3.13 describes a plausible relationship among all variables. Variable X_3 has a direct causal impact on the outcome S while the path between X_1 and S is both direct and also mediated by X_2, X_3 and class ($\hat{C}$), with X_1 possibly having a higher direct causal impact on class ($\hat{C}$) than its direct impact on the outcome (S). X_2 has a causal impact on both class and survival. This hypothetical scenario justifies the position of X_1 and X_3 in a life course framework, where distal exposures (e.g. X_1) tend to have a weaker direct causal impact on the outcome while proximal exposures, such as X_3 , have a stronger direct causal impact on the outcome, but the former nevertheless still contributes to population heterogeneity.

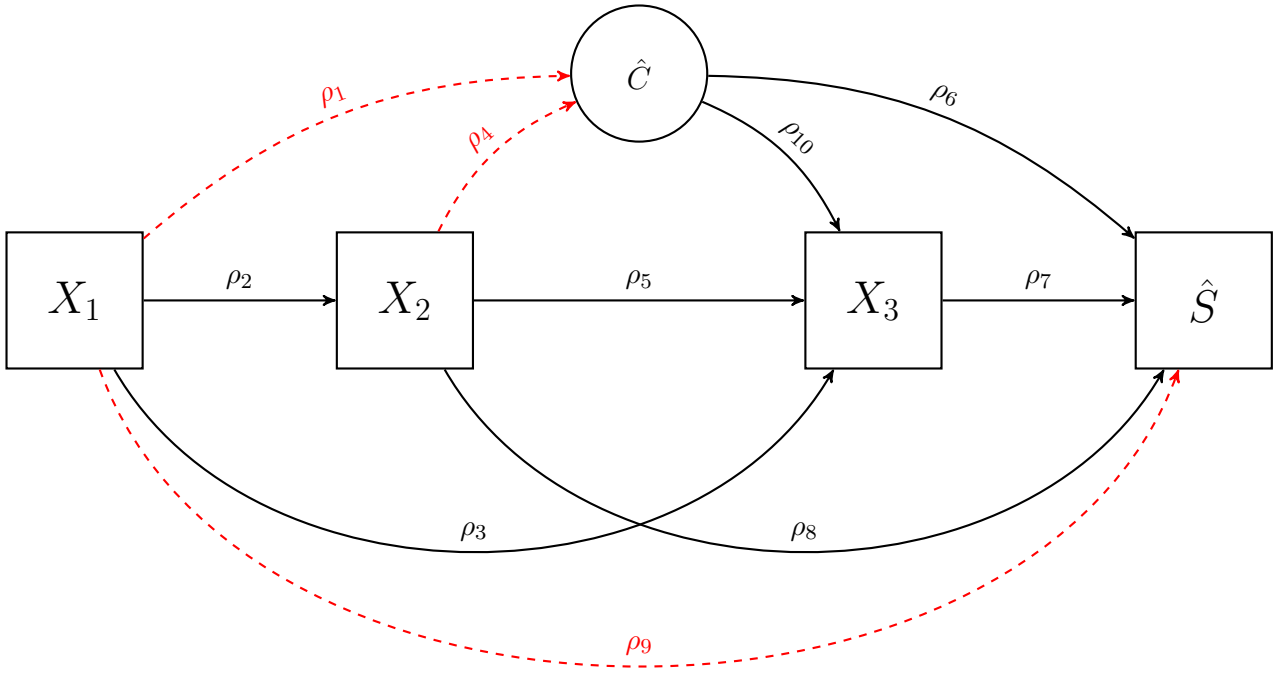


Figure 3.13: A hypothetical temporal-causal diagram depicting the causal relationships amongst three predictors (X_1, X_2 and X_3), one latent class (C), and the outcome (death/survival; S) in a simulated observational setting where preceding covariates act as potential causes of all subsequent variables, including class and/or death/survival.

These simulations are designed to mimic an observational study setting so that we can investigate the role of lifecourse exposures in predicting both class-membership and the health outcome (e.g. survival). Informed by real life studies, we let $\mathbf{X} = \{X_1, X_2, X_3, \hat{S}, \hat{C}\}$ be a vector of predictors drawn from the multivariate normal distribution. The class C variable was generated by arbitrarily splitting a normally distributed variable, $\hat{C}$ into two groups such that 70% in class 1 and 30% in class 2–as approximately observed in the example dataset on patients with coronary heart failure disease. To mimic this dataset, we transformed a

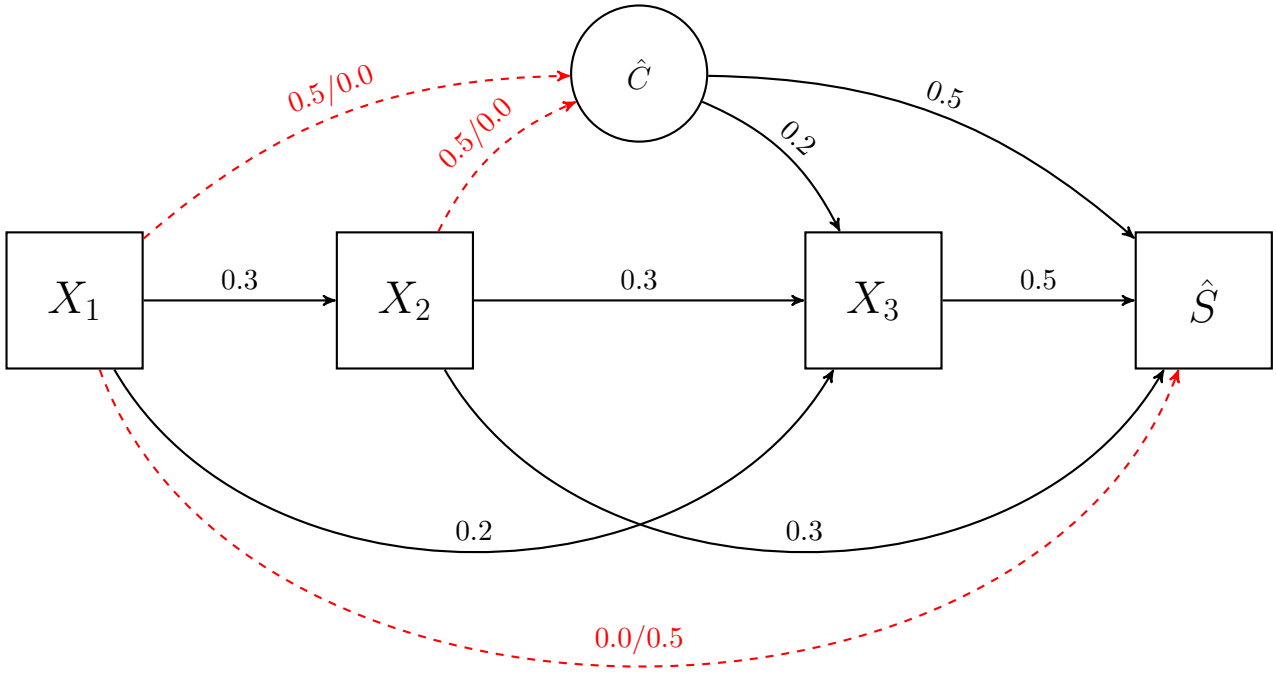


Figure 3.14: In the temporal-causal diagram of Figure 3.13, path coefficients are either constant or summarised for all three scenarios considered. The key paths that mediate distal and intermediate predictor influence to the outcome via population heterogeneity are given dotted lines.

normally distributed variable $\hat{S}$ into a survival outcome S to represent the length of followup for patients with CHF disease.

There are many distributions that are used to model survival data. The exponential distribution is the common approach which is commonly adopted. Other distributions such as the Weibull and gamma, Gompertz, lognormal, and log-logistic are useful alternatives. The Weibull distribution (i.e. $\mathcal{W}(\eta, \lambda)$) characterised by two parameters, η (shape parameter) and λ (scale parameter) has been recommended in some previous studies(Lee and Go, 1997). The time to event, S was

initially drawn from a normal (i.e $\mathcal{N}(\mu, \sigma^2)$) and converted to an exponential random variable with survival time ranging from 0 to 25 years.

Assumptions for the Data generation mechanism in Figure 3.13

It is important to realise that initially naïve assumptions are being made about the causal graph in order to arrive ultimately at the desired data structure. This DGM makes three plausible assumptions on the basis of the relative temporal position of each variable (Ellison, 2021):

- Only preceding variables could act as potential causes of subsequent variables. This is supported by Pearl and Verma (Pearl and Verma, 1995) who argued that temporal ordering of variables is essential for defining causation and that it may also help to distinguish causal from other types of associations.

- Any preceding variable could act as a potential cause of all subsequent variables. For example, early life nutritional health status could be a potential cause of obesity later in life.

- The strength of the causal relationships and their associated path coefficients are dependent upon the variables concerned and the specific context(s) involved.

It is important to note that he pre-transformed simulated data are drawn from a multivariate normal distribution. The path coefficients in the causal DAG will not therefore represent the true relationship in the post-transformed simulated

data, which is why there is an iterative process in arriving at the correct path coefficients involving variables in the DAG that will need to be transformed.

To simulate data in R, we created a hypothetical diagram using an R package called *daggity* (Textor et al., 2016) with arrows depicting causal relationships between covariates (X_1, X_2, X_3) and outcomes ($\hat{C}$ and $\hat{S}$) in an observational study setting. Data were simulated using a *simulateSEM* function within the *daggity* package. The *simulateSEM* function interprets a causal path diagram using Wright's rules. Both the class variable ($\hat{C}$) and survival ($\hat{S}$) are initially simulated as multivariate normal, with $\hat{C}$ transformed to binary C(i.e. representing latent subgroups of patients) and $\hat{S}$ transformed to S.

3.4.1 An illustration of Wright's Rules: Application to a DAG depicting a temporal order of variables

In this DAG, X_1,X_2 and X_3 depict a temporal order of variables. These variables are used to predict the pre-transformed patients' survival ($\hat{S}$) and class-membership ($\hat{C}$).

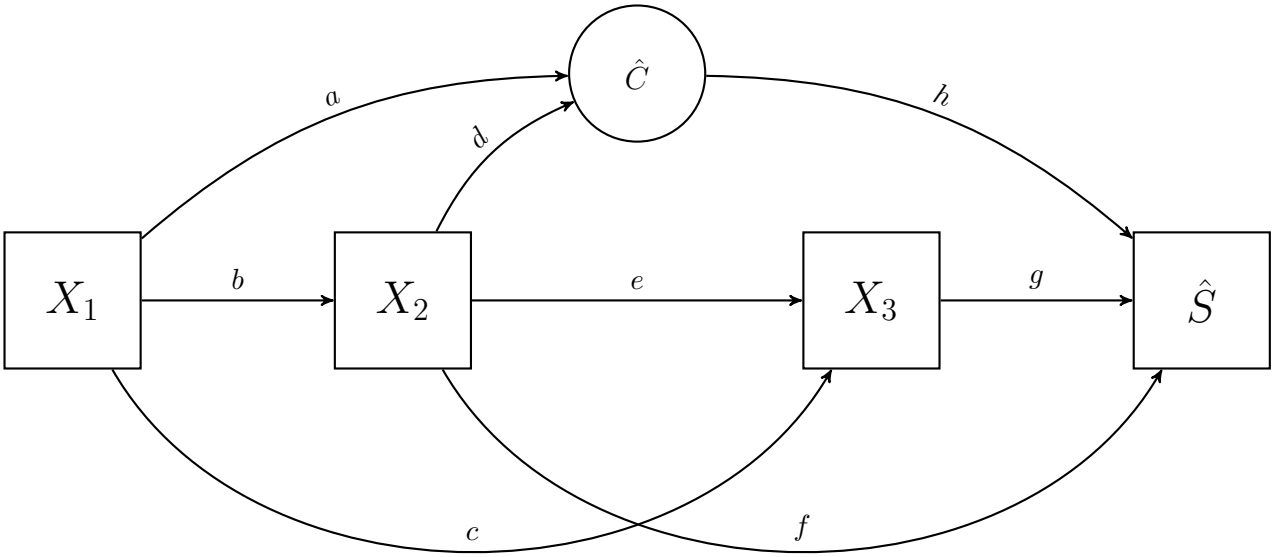


Figure 3.15: A hypothetical causal diagram for an observational study setting.

The path coefficients are represented by a, b, c, d, e, f, g, h as shown in Fig 3.16. Suffice to note that b, c and e represent correlations amongst independent variables. To generate a covariance structure for Fig 3.16, we first of all decompose the correlation structure amongst the five variables where $\hat{C}$ and $\hat{S}$ are outcome variables.

There are two direct effects to $\hat{C}$, the first one is from X_1 to $\hat{C}$ and another one from X_2 to $\hat{C}$. X_1 indirectly causes $\hat{C}$ mediated by X_2. similarly, X_2 causes C through X_1. X_3 has no direct causal impact on C, but it indirectly impacts C through X_2 and X_1. Similarly, there are two direct effects to $\hat{S}$, the first one is from X_2 to $\hat{S}$ and another one from X_3 to S. X_1 indirectly causes S mediated by $\hat{C}$. X_1 also indirectly causes $\hat{S}$ mediated by X_2 and X_3. Additionally, X_1 indirectly causes $\hat{S}$ mediated by X_2 and $\hat{C}$. X_2 indirectly causes $\hat{S}$ through X_3 and $\hat{C}$. Additionally, X_2 indirectly causes $\hat{S}$ mediated by X_1 and $\hat{C}$. X_3 has an

indirect causal impact on $\hat{S}$ through X_2 and $\hat{C}$. X_3 has another indirect causal impact on $\hat{S}$ through X_1 and $\hat{C}$ as well as through X_2.

Using wright's rules outlined on section 2.5.1, the correlations among these five variables can be decomposed as follows: The correlation between X_1 and X_2 is represented by b. The correlation between X_2 and X_3 is represented by e while the correlation between X_1 and X_3 is represented by c. Therefore, we can write $r_{12} = b$, $r_{23} = e$ and $r_{13} = c$ respectively. The correlation between X_1 and $\hat{C}$ is given by $r_{cx_1} = a+bd$. The correlation between X_2 and $\hat{C}$ is given by $r_{cx_1} = d+ab$. The correlation between X_3 and $\hat{C}$ is given by $r_{cx_1} = de + ac$. Similarly, the correlation between X_1 and $\hat{S}$ is given by $r_{sx_1} = ah + bdh + cg + bf$. The correlation between X_2 and $\hat{S}$ is given by $r_{sx_2} = f + eg + dh + bdh$. Finally, the correlation between X_3 and $\hat{S}$ is given by $r_{sx_3} = g + edh + cah + ef$.

In summary, we have a correlation matrix defined as follows:

$$\Omega = \begin{bmatrix} 1 & & & & \\ h & 1 & & & \\ a+bd & ah+bdh+cg+bf & 1 & & \\ d+ab & f+eg+dh+bdh & b & 1 & \\ de+ac & g+edh+cah+ef & c & e & 1 \end{bmatrix}$$

To illustrate how this works, assume that the the observed correlations are as follows: $r_{12} = 0.02$, $r_{13} = 0.01$ and $r_{23} = 0.02$. Let us further assume that the path coefficients are given as follows, $a = 0.6, d = 0.2, f = 0.4, g = 0.7, h = 0.5$.

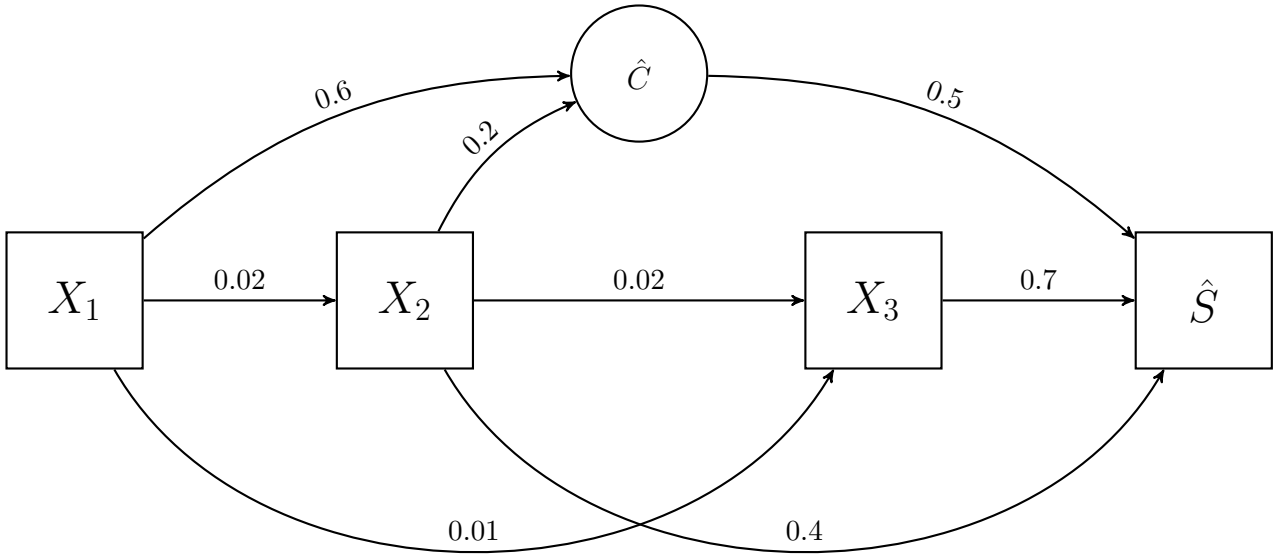


Figure 3.16: A hypothetical causal diagram for an observational study setting.

Then the correlation structure is given by:

$$\Omega' = \begin{bmatrix} 1.00 & & & & \\ 0.50 & 1.00 & & & \\ 0.60 & 0.32 & 1.00 & & \\ 0.21 & 0.52 & 0.02 & 1.00 & \\ 0.01 & 0.71 & 0.01 & 0.02 & 1.00 \end{bmatrix}$$

3.5 Chapter Summary

The standard simulation process involves simulating data that follows a particular covariance structure. The challenge with this approach is that sometimes the range of simulations considered might fail to capture the realistic potential scenarios as dictated by the underlying data generating mechanisms, which may not be apparent by simply specifying the resultant covariance structure. To simulate data that reflect the underlying truth in more complex real-world settings, we must inspect the temporal ordering of variables to consider their causal interplay and thus examine how this may affect any future outcomes – where this is not known a *priori*, as with lifecourse data, then all potential options for a dataset must be anticipated.

We have introduced the process of how to generate simulations that respect the causal data generation processes operating a longitudinal setting, to reflect the underlying realistic scenarios we hypothesise, that lie behind the observed data. We have explored different contexts to illustrate the benefits of simulating the data generating process over merely simulating the consequential covariance structure, but these illustrations will have a practical utility in the next two chapters where we evaluate the consequence of having a range of data generating processes when exploring a particular method. Adopting a DAG based simulation provides greater complexity in the simulation to capture realistic problems and features of real data. For instance, in the prediction of change example, we drew attention to the importance of recognising composite measures and deconstructing these into their parent components; in the prediction of longitudinal outcomes, we outlined the different roles of predictors at various stages of life, and how this can

lead to an understanding of latent inherent heterogeneity within a dataset. This highlights the benefits that stems from including a temporal component in our thinking behind causal understanding of the data generation and the explicit use of DAGs to exploit this in informing simulations that would otherwise be totally lost in the simpler standard methods of beginning with a postulated covariance matrix.

While we acknowledge that simulating with respect to a causal structure is not novel, substantial novelty exists in the simulations undertaken in this thesis around the assumed Data Generating Mechanisms (DGMs) adopted to reflect natural process change in an outcome over time (e.g. follow-up *change*, opposed to *change-scores*) and the lifecourse consequences of enigmatic variation (i.e. small but frequent random nudges in one direction or another) that generates population heterogeneity in all outcomes over time. Heterogeneity is extremely common in health data, yet very poorly or inadequately addressed explicitly. To our knowledge, none of these DGMs have been previously examined with respect to prediction (i.e. through the lens of causal inference with a view to improve prediction). Others have shown the utility of causal insight for improved prediction e.g. (Piccininni et al., 2020; Richens et al., 2020) but we specifically examine two scenarios in depth (i.e. the analysis of follow-up change versus change-scores, and the later-life outcome (survival) in response to substantially varied lives within heterogeneous populations). No previous work has consolidated either idea in a comprehensible illustration that addresses the prediction challenges we highlight, to guide researchers who may be interested in doing similar work to improve prediction models and their reliability for different contexts.

Chapter 4

Predicting change-scores and follow-up outcomes in an observational study setting; evaluation and recommendations

In this Chapter, we discuss the first illustration of our simulation process described in Chapter 3. We begin by introducing the concept of *change* before introducing an illustrative example where we evaluate prediction models under different scenarios. The ultimate goal in this chapter is to establish ways of improving the prediction of *change* given a number of complex circumstances within an observation study setting.

4.1 Introduction

Longitudinal studies examining the relationship between baseline exposure(s) and the subsequent *change* in health status or the putative *outcome* of interest are common in epidemiological research. For instance, suppose that in an epidemiological study, a dependent outcome, Y is measured at baseline, Y_0 and follow-up, Y_1. Assuming that we are interested in assessing the relationship between an exposure variable X , measured at baseline (hence depicted X_0) and the changes that arise in the outcome Y. To assess this relationship between X and Y, there are two proposed regression method strategies that are commonly applied, namely the change score analysis and the regressor method known as the analysis of covariance method (Allison, 1990; Senn, 2006).

The change score analysis involves regressing the outcome-change score ($\Delta Y = Y_1 - Y_0$) on the baseline exposure (X_0) while ANCOVA is where the follow-up outcome, Y_1 is regressed on the baseline exposure, X_0 while adjusting for the baseline outcome Y_0 (Tennant et al., 2021a). It has been shown mathematically that these two approaches are equivalent and both yield the same model coefficients and standard errors (Werts and Linn, 1970).

Even though these methods have been widely used to analyse change or followup measurements, there has been very little evaluation of their role in prediction. Applications of these methods for prediction of change in a longitudinal outcome have not been widely explored. The main focus has been on the interpretations of the model coefficients and not on the assessment of the outcome predictions, e.g. how model performance and parsimony might differ between strategies.

This chapter aims at applying the change score and regressor method to predict the change or follow-up outcomes to evaluate potential differences in predictor variable selection and prediction performance under a range of controlled (and therefore known) conditions. An illustrative example that follows the ADEMP structure of reporting simulation studies as described by Morris et al (Morris et al., 2019) is adopted to aid in understanding and facilitate the evaluation of different plausible scenarios that may be encountered in real epidemiological studies.

4.2 An illustrative example

Suppose that Y is an outcome variable with two measurements , Y_0 (i.e. measured at *baseline* meaning that the outcome value is obtained at the beginning of the study) and Y_1 (i.e. measured at *follow-up* meaning that the outcome value is obtained at the end of the study period). Let X_0 and U_0 be the exposure and competing exposure variables for the outcome Y. For example, suppose weight at baseline is denoted Y_0 and the weight after one year is denoted Y_1. The predictors X_0 and U_0 could represent age and BMI respectively. Using these variables, we now describe a simulation study to evaluate the two proposed regression strategies (i.e. the change score analysis and the regressor method) in terms of prediction performance and variable selection.

4.2.1 Aims

The aim of the study is to:

1. Evaluate the impact of forcibly including the baseline, Y_0 as one of the

predictors of either change score, ΔY, or follow-up, Y_1 .

2. Evaluate the impact of forcibly excluding the baseline, Y_0 as one of the predictors of either change score, ΔY, or follow-up, Y.

3. Assess the implications of allowing the prediction model algorithm adopted to select from candidate predictors, Y_0, X_0 and U_0 while predicting the change-score, ΔY, or follow-up, Y.

4. Assess the differences in root mean square error of option (3) within the test data between the change-score model and the follow-up model.

4.2.2 Data generating mechanisms

We considered twelve directed acyclic graphs (DAGs) depicting several plausible causal scenarios for simulation, as shown in Figures 3.1 - 3.12. These fall into three broad causal contexts:

In the first context we assume orthogonality between the baseline exposure and the baseline outcome measurements, meaning that the two are uncorrelated (as within a randomised trial setting)-as such, there is no arrow connecting X_0 and Y_0 in all DAGs under this assumption.

In the second context, it was assumed that X_0 confounds Y_0, meaning that two are correlated. There is thus an arrow from X_0 to Y_0 for all DAGs under this assumption.

In the third context, the assumption is that X_0 mediates Y_0, meaning that again the two are correlated- this time, however, there is an indirect route from Y_0 to Y_1 through X_0. For each context, different assumptions and parameter specifications

were considered for the data generating mechanism, as shown in the tables 4.1-4.3. It is these correlations and the specified path coefficients that were used to generate a positive definite covariance matrix for each scenario which formed the *solution space*. For each scenario, 100 simulated datasets were generated, each with 1000 observations and four variables, Y_0, Y_1, X_0 and U_0 where Y_0 is the baseline measurement for the outcome, Y_1 is the follow-up measurement, X_0 and U_0 are assumed to be the baseline exposure and competing exposure variables respectively. All variables were sampled from a multivariate normal distribution.

Description of data generating mechanisms

Firstly, we assume that X_0 and Y_0 are orthogonal -this would be a typical of the situation for a randomised control trial, for instance, where the randomisation ensure orthogonality. Below is a summary of the parameters.

Coding	Correlation	Parameters	Description
		Scenario 1: X_0 and Y_0 are orthogonal & no U_0 confounding	
$\rho_{X_0Y_0}$	0.0	1	X_0 and Y_0 are orthogonal, no correlation
$\rho_{U_0Y_0}$, $\rho_{U_0X_0}$	0.0	1	No U_0 confounding
$\rho_{Y_0Y_1}$	$\{0.05, 0.10, \ldots, 0.95\}$	19	A positive correlation between Y_0 and Y_1
$\rho_{X_0Y_1}$	$\{-0.95, -0.90, \ldots, 0.95\}$	38	Both positive and negative correlation between X_0 and Y_1.
$\rho_{U_0Y_1}$	$\{-0.5, 0.5\}$	2	Either a correlation of -0.5 or 0.5 between U_0 and Y_1
		Scenario 2: X_0 and Y_0 are orthogonal & U_0 confounds X_0	
$\rho_{X_0Y_0}$	0.0	1	X_0 and Y_0 are orthogonal
$\rho_{U_0Y_0}$	0.0	1	No U_0 confounding
$\rho_{Y_0Y_1}$	$\{0.05, 0.10, \ldots, 0.95\}$	19	Baseline outcome serial correlation path coefficient
$\rho_{X_0Y_1}$	$\{-0.95, -0.90, \ldots, 0.95\}$	38	Main covariate effect size path coefficient
$\rho_{U_0X_0}$, $\rho_{U_0Y_1}$	$\{-0.5, 0.5\}$	2	Either a correlation of -0.5 or 0.5 between U_0 and X_0 as well as between U_0 and Y_1
		Scenario 3: X_0 and Y_0 are orthogonal & U_0 confounds Y_0	
$\rho_{X_0Y_0}$	0.0	1	X_0 and Y_0 are orthogonal
$\rho_{U_0X_0}$	0.0	1	No U_0 confounding
$\rho_{Y_0Y_1}$	$\{0.05, 0.10, \ldots, 0.95\}$	19	A positive correlation between Y_0 and Y_1
$\rho_{X_0Y_1}$	$\{-0.95, -0.90, \ldots, 0.95\}$	38	Main covariate effect size path coefficient
$\rho_{U_0Y_0}$, $\rho_{U_0Y_1}$	$\{-0.5, 0.5\}$	2	Either a correlation of -0.5 or 0.5 between U_0 and Y_0 as well as between U_0 and Y_1
		Scenario 4: X_0 and Y_0 are orthogonal & no U_0 mediates Y_0	
$\rho_{X_0Y_0}$	0.0	1	X_0 and Y_0 are orthogonal
$\rho_{U_0X_0}$	0.0	1	No U_0 confounding
$\rho_{Y_0Y_1}$	$\{0.05, 0.10, \ldots, 0.95\}$	19	A positive correlation between Y_0 and Y_1
$\rho_{X_0Y_1}$	$\{-0.95, -0.90, \ldots, 0.95\}$	38	Main covariate effect size path coefficient
$\rho_{U_0Y_0}$, $\rho_{U_0Y_1}$	$\{-0.5, 0.5\}$	2	Either a correlation of -0.5 or 0.5 between U_0 and Y_0 as well as between U_0 and Y_1

Table 4.1: Summary of the parameters for the Scenarios in which X_0 and Y_0 are orthogonal

Secondly, we assume that X_0 confounds Y_0 -this would represent a situation where X_0 crystallises before Y_0. The following are the parameters used to simulate data.

Coding	Correlation	Parameters	Description
		Scenario 1: X_0 confounds Y_0 & no U_0 confounding	
$\rho_{U_0Y_0}$, $\rho_{U_0X_0}$	0.0	1	No U_0 confounding
$\rho_{Y_0Y_1}$	$\{0.05, 0.10, \ldots, 0.95\}$	19	A positive correlation between Y_0 and Y_1
$\rho_{X_0Y_1}$	$\{-0.95, -0.90, \ldots, 0.95\}$	38	Main covariate effect size path coefficient
$\rho_{U_0Y_1}$, $\rho_{X_0Y_0}$	$\{-0.5, 0.5\}$	2	Either a correlation of -0.5 or 0.5 between X_0 and Y_0 as well as between U_0 and Y_1
		Scenario 2: X_0 confounds Y_0 & U_0 confounds X_0	
$\rho_{U_0Y_0}$	0.0	1	No U_0 confounding
$\rho_{Y_0Y_1}$	$\{0.05, 0.10, \ldots, 0.95\}$	19	A positive correlation between Y_0 and Y_1
$\rho_{X_0Y_1}$	$\{-0.95, -0.90, \ldots, 0.95\}$	38	Main covariate effect size path coefficient
$\rho_{U_0X_0}$, $\rho_{U_0Y_1}$, $\rho_{X_0Y_0}$	$\{-0.5, 0.5\}$	2	Either a correlation of -0.5 or 0.5 between U_0 and X_0, between U_0 and Y_1 as well as between X_0 and Y_0
		Scenario 3: X_0 confounds Y_0 & U_0 confounds Y_0	
$\rho_{U_0X_0}$	0.0	1	No U_0 confounding
$\rho_{Y_0Y_1}$	$\{0.05, 0.10, \ldots, 0.95\}$	19	A positive correlation between Y_0 and Y_1
$\rho_{X_0Y_1}$	$\{-0.95, -0.90, \ldots, 0.95\}$	38	Main covariate effect size path coefficient
$\rho_{U_0Y_0}$, $\rho_{U_0Y_1}$, $\rho_{X_0Y_0}$	$\{-0.5, 0.5\}$	2	Either a correlation of -0.5 or 0.5 between U_0 and Y_0, between U_0 and Y_1 as well as between X_0 and Y_0
		Scenario 4: X_0 confounds Y_0 & no U_0 confounds X_0 & Y_0	
$\rho_{Y_0Y_1}$	$\{0.05, 0.10, \ldots, 0.95\}$	19	A positive correlation between Y_0 and Y_1
$\rho_{X_0Y_1}$	$\{-0.95, -0.90, \ldots, 0.95\}$	38	Main covariate effect size path coefficient
$\rho_{U_0Y_0}$, $\rho_{U_0Y_1}$, $\rho_{U_0X_0}$,$\rho_{X_0Y_0}$	$\{-0.5, 0.5\}$	2	Either a correlation of -0.5 or 0.5 between U_0 and Y_0, between U_0 and Y_1, between U_0 and X_0 as well as between X_0 and Y_0

Table 4.2: Summary of the parameters for the Scenarios in which X_0 confounds Y_0.

Lastly, we assume that X_0 mediates Y_0 -this would represent a situation where Y_0 crystallises before X_0, for instance. The following are the parameters used to simulate data.

4.2.3 Estimand

Our targeted estimand is the change-score which would represent a change in blood pressure after an intervention in an observational study setting.

Coding	Correlation	Parameters	Description
		Scenario 1: X_0 mediates Y_0 & no U_0 confounding	
$\rho_{U_0Y_0}$, $\rho_{U_0X_0}$	0.0	1	No U_0 confounding
$\rho_{Y_0Y_1}$	$\{0.05, 0.10, \ldots, 0.95\}$	19	A positive correlation between Y_0 and Y_1
$\rho_{X_0Y_1}$	$\{-0.95, -0.90, \ldots, 0.95\}$	38	Main covariate effect size path coefficient
$\rho_{U_0Y_1}, rho_{Y_0X_0}$	$\{-0.5, 0.5\}$	2	Either a correlation of -0.5 or 0.5 between U_0 and Y_1 as well as between Y_0 and X_0
		Scenario 2: X_0 mediates Y_0 & U_0 confounds X_0	
$\rho_{U_0Y_0}$	0.0	1	No U_0 confounding
$\rho_{Y_0Y_1}$	$\{0.05, 0.10, \ldots, 0.95\}$	19	A positive correlation between Y_0 and Y_1
$\rho_{X_0Y_1}$	$\{-0.95, -0.90, \ldots, 0.95\}$	38	Main covariate effect size path coefficient
$\rho_{U_0X_0}, \rho_{U_0Y_1}, \rho_{Y_0X_0}$	$\{-0.5, 0.5\}$	2	Either a correlation of -0.5 or 0.5 between U_0 and Y_1, between U_0 and X_0 as well as between Y_0 and X_0
		Scenario 3: X_0 mediates Y_0 & U_0 confounds Y_0	
$\rho_{U_0X_0}$	0.0	1	No U_0 confounding
$\rho_{Y_0Y_1}$	$\{0.05, 0.10, \ldots, 0.95\}$	19	A positive correlation between Y_0 and Y_1
$\rho_{X_0Y_1}$	$\{-0.95, -0.90, \ldots, 0.95\}$	38	Main covariate effect size path coefficient
$\rho_{U_0Y_0}, \rho_{U_0Y_1}, \rho_{Y_0X_0}$	$\{-0.5, 0.5\}$	2	Either a correlation of -0.5 or 0.5 between U_0 and Y_0, between U_0 and Y_1 as well as between Y_0 and X_0
		Scenario 4: X_0 mediates Y_0 & no U_0 confounds X_0 & Y_0	
$\rho_{Y_0Y_1}$	$\{0.05, 0.10, \ldots, 0.95\}$	19	A positive correlation between Y_0 and Y_1
$\rho_{X_0Y_1}$	$\{-0.95, -0.90, \ldots, 0.95\}$	38	Main covariate effect size path coefficient
$\rho_{U_0Y_0}, \rho_{U_0Y_1}, \rho_{U_0X_0}, \rho_{X_0Y_0}$	$\{-0.5, 0.5\}$	2	Either a correlation of -0.5 or 0.5 between U_0 and Y_0, between U_0 and Y_1, between U_0 and X_0 as well as between X_0 and Y_0

Table 4.3: Summary of the parameters for the Scenarios in which X_0 mediates Y_0

4.2.4 Methods

Each simulated dataset is analysed using the two linear regression models as described below. The first regression model is constructed for the outcome Y_1 regressed on candidate covariates, X_0, U_0 and Y_0 with predictors retained in the best model selected according to the Bayesian Information Criterion (BIC) adopting the all subsets regression method which runs regression models on all possible permutations of candidate predictors (Nimon and Oswald, 2013). Bayesian Information Criterion (BIC) and Alkaike Information Criteria (AIC) have been widely used for model selection in linear regression models (Lee et al., 2014; Li

and Nyholt, 2001). The BIC is calculated as follows:

$$BIC = -2\log(L) + K\log(N) \tag{4.1}$$

while AIC is calculated as follows

$$AIC = -2\log(L) + 2K \tag{4.2}$$

where L is the maximum likelihood, K the number of parameters to be estimated in the model, and N the sample size. In this analysis we deployed BIC to select the best model from a list of competing models because it is the most parsimonious and thus avoids overfitting more than the AIC. The lower the BIC, the better the model.

The first regression model was constructed as follows:

$$Y_1 = \beta_0 + \beta_1 Y_0 + \beta_2 U_0 + \beta_3 X_0 + \epsilon_1 \tag{4.3}$$

The second regression model uses the constructed change score, ΔY, as the outcome and this is regressed on candidate covariates, X_0, U_0 and Y_0 with predictors retained in the best model according to the BIC when adopting the all subsets regression method, as before.

$$\Delta Y = \beta_0 + \beta_1 Y_0 + \beta_2 U_0 + \beta_3 X_0 + \epsilon_2 \tag{4.4}$$

Training Process

Each simulated data comprised four variables Y_0, Y_1, X_0 and U_0 with a sample size of 1000. All the variables were drawn from a multivariate normal distribution. A change-score variable was calculated by subtracting the baseline measurement from the follow-up measurement (i.e. $\Delta Y = Y_1 - Y_0$). Each dataset was randomly split into the training and datasets. The training dataset comprised 70% of the original dataset (i.e. n=700) and testing dataset comprised the remaining 30% of the original data (i.e. n=300). Two models were considered for training in each case, the first being the ANCOVA model with the follow-up variable (i.e. Y_1) as its outcome followed by the change-score model with the change variable (i.e. $\Delta Y = Y_1 - Y_0$) as its outcome.

To address the four objectives for this chapter, three types of models were explored for each outcome. The three scenarios that were of interest were:

1. Forcibly including the baseline as a predictor while selecting from the two exposure variables based on the best BIC.

2. Selecting predictors from the baseline outcome and two exposure variables based on the best BIC.

3. Forcibly excluding the baseline as a predictor while selecting from the two exposure variables based on the best BIC.

Each scenario involved two models, one for each choice of outcome. In the first scenario, where Y_0 was forcibly included, some models retained both candidate predictors while others dropped one or both candidate predictors, depending upon the strength of the joint associations between the predictors and the outcome.

The following are the four possible models that were explored for an ANCOVA model with Y_1 as an outcome.

1. Model 1: $Y_1 \sim \beta_0 + \beta_1 Y_0 + \epsilon_1$ - Both X_0 and U_0 dropped.
2. Model 2: $Y_1 \sim \beta_0 + \beta_1 Y_0 + \beta_2 X_0 + \epsilon_1$ - U_0 dropped.
3. Model 3: $Y_1 \sim \beta_0 + \beta_1 Y_0 + \beta_2 U_0 + \epsilon_1$ - X_0 dropped.
4. Model 4: $Y_1 \sim \beta_0 + \beta_1 Y_0 + \beta_2 U_0 + \beta_3 X_0 + \epsilon_1$ - Both X_0 and U_0 retained.

The same set of models were explored for the change-score outcome where $\Delta Y = Y_1 - Y_0$ defined as the outcome variable.

In the second scenario, the predictors for the two models were selected from a set of three candidate predictors, Y_0, U_0, X_0. Some models retained the baseline, Y_0 while other models did not, depending on the underlying covariance structure. The following are the eight possible models that were explored for a model with Y_1 as an outcome.

1. Model 1: $Y_1 \sim \beta_0 + \epsilon_1$ - X_0, Y_0 and U_0 dropped.
2. Model 2: $Y_1 \sim \beta_0 + \beta_1 Y_0 + \epsilon_1$ - Both X_0 and U_0 dropped.
3. Model 3: $Y_1 \sim \beta_0 + \beta_1 U_0 + \epsilon_1$ - Both X_0 and Y_0 dropped.
4. Model 4: $Y_1 \sim \beta_0 + \beta_1 X_0 + \epsilon_1$ - Both Y_0 and U_0 dropped.
5. Model 5: $Y_1 \sim \beta_0 + \beta_1 Y_0 + \beta_2 X_0 + \epsilon_1$ - U_0 dropped.
6. Model 6: $Y_1 \sim \beta_0 + \beta_1 Y_0 + \beta_2 U_0 + \epsilon_1$ - X_0 dropped.

7. Model 7: $Y_1 \sim \beta_0 + \beta_1 U_0 + \beta_2 X_0 + \epsilon_1$ - Y_0 dropped.

8. Model 8: $Y_1 \sim \beta_0 + \beta_1 Y_0 + \beta_2 X_0 + \beta_3 U_0 + \epsilon_1$ - all three predictors retained.

The same set of models were also explored for the change score outcome with $\Delta Y = Y_1 - Y_0$ defined as the outcome variable.

Lastly, the baseline measurement variable was forcibly ignored when modelling the two outcomes. Models were allowed to select predictors from the set of candidate predictors; U_0 and X_0. The following are possible models that were explored for a model with Y_1 as an outcome.

1. Model 1: $Y_1 \sim \beta_0 + \epsilon_1$ - X_0 and U_0 dropped.

2. Model 2: $Y_1 \sim \beta_0 + \beta_1 X_0 + \epsilon_1$ - U_0 dropped.

3. Model 3: $Y_1 \sim \beta_0 + \beta_1 U_0 + \epsilon_1$ - X_0 dropped.

4. Model 4: $Y_1 \sim \beta_0 + \beta_1 X_0 + \beta_1 U_0 + \epsilon_1$ - Both X_0 and U_0 have been retained.

The same set of models were explored for the change-score outcome with $\Delta Y = Y_1 - Y_0$ defined as the outcome variable.

4.2.5 Performance measures

The root mean square error (RMSE) is typically used as a standard method for model performance evaluation in many studies (Brassington, 2017; Chai and Draxler, 2014). The RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.5}$$

All models were trained using 70% of the dataset and the BIC's calculated. The best model (i.e. the model with the lowest BIC value) was selected and tested using 30% of the sample. To assess the performance of the predictions, the RMSE was calculated. The model with the lowest RMSE is the best model.

4.3 Summary of results

Table 4.4 provides a summary of the results from the models with predictors selected from the choice amongst X_0 and U_0 only with Y_0 forcibly included as default, for either Y_1 as the outcome or $\Delta Y = Y_1 - Y_0$ as the outcome. The first column shows the correlations between the baseline and the follow-up measurements. The second column shows the correlation between the exposure variable and the follow-up measurement. The third and fourth column shows the correlations between the competing exposure variable and the follow-up measurement as well as between the competing exposure variable and the baseline outcome measurements. The fifth column shows the correlation between the baseline exposure and baseline outcome measurements. The sixth column shows the correlation between the baseline exposure variable and the competing exposure. The last two columns indicate the predictors that are retained in each model for both outcomes based on different solution spaces where the implied covariance matrix is positive definite. When the baseline outcome measurement, Y_0, is forcibly included as a predictor for the scenarios evaluated, there is never any difference between predicting the change-score outcome, ΔY or the ANCOVA followup outcome, Y_1 as expected, since we know that once Y_0 is conditioned on for the outcome ΔY, this is mathematically equivalent to modelling the outcome Y_1. Inspecting the

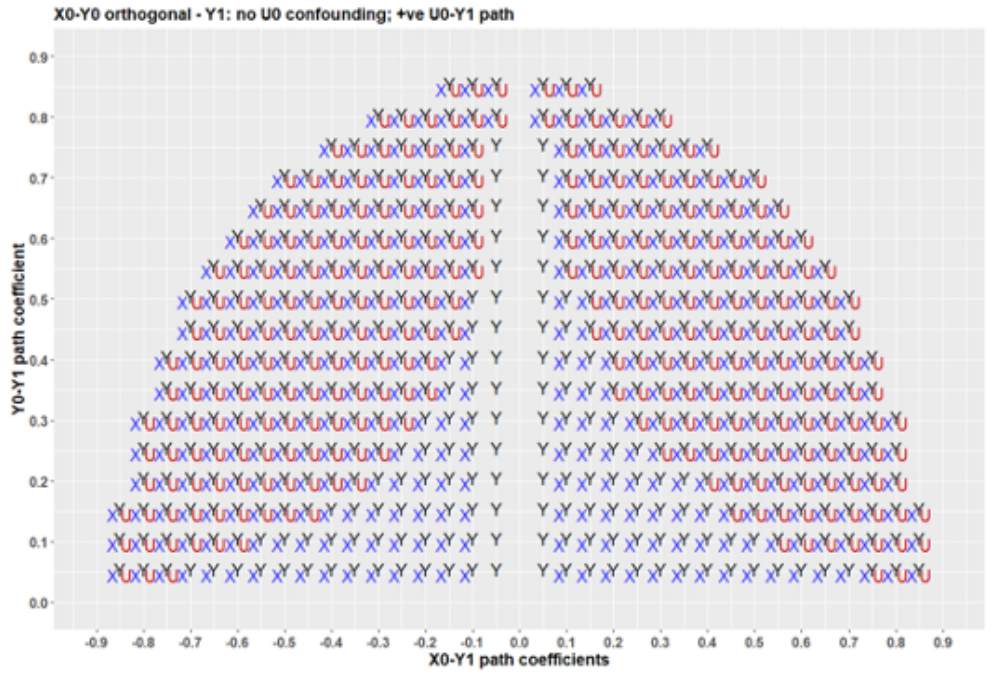
graphs in Figure 4.1, we notice that when Y_0 is forcibly included in the model as a default variable for the two models (either Y_1 as the outcome or $\Delta Y = Y_1 - Y_0$ as the outcome), the same predictors are retained in both cases. The first graph (a) is the solution space for the model predicting Y_1 while the second graph (b) is the solution space for the model predicting change score, ΔY. The two graphs are clearly the same. There are no differences between the predictors retained in both models suggesting that the two models are equivalent.

Force Y_0

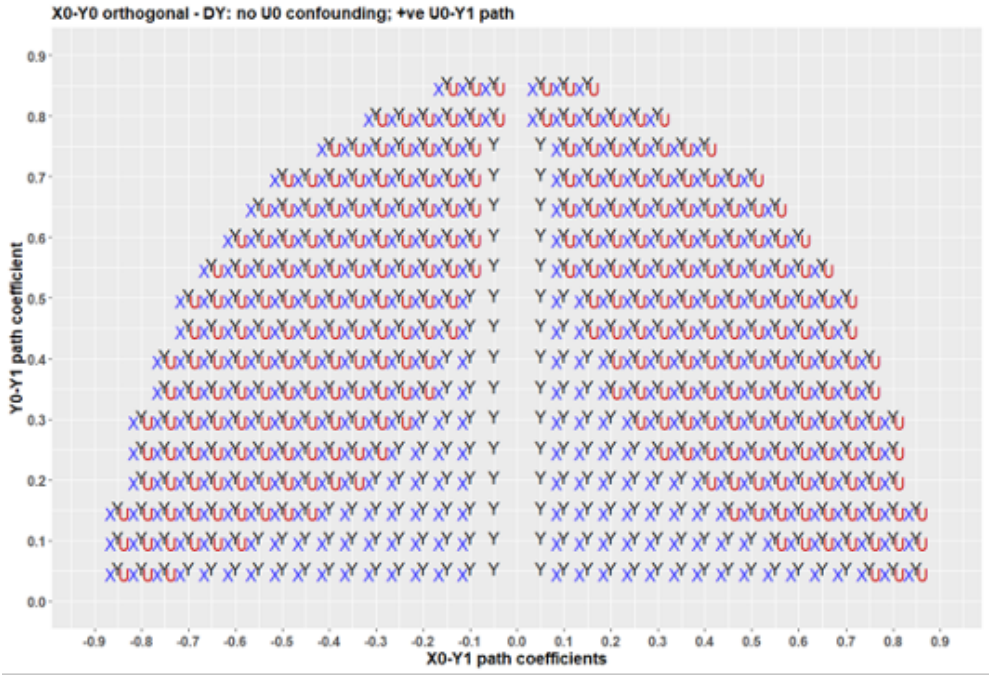
$\rho_{Y_0Y_1}$	$\rho_{X_0Y_1}$	$\rho_{U_0Y_1}$	$\rho_{U_0Y_0}$	$\rho_{X_0Y_0}$	$\rho_{U_0X_0}$	pred Y_1	pred ΔY
0.05	−0.70	−0.5	−0.5	−0.5	−0.5	$\{X_0, U_0\}$	$\{X_0, U_0\}$
0.05	−0.95	−0.5	0.0	−0.5	−0.5	$\{X_0, U_0\}$	$\{X_0, U_0\}$
0.05	0.05	0.5	0.5	−0.5	0.0	$\{U_0\}$	$\{U_0\}$
0.95	0.95	−0.5	0.0	−0.5	0.5	$\{X_0, U_0\}$	$\{X_0, U_0\}$
0.25	0.4	−0.5	0.0	0.5	−0.5	$\{X_0\}$	$\{X_0\}$
0.6	0.65	−0.5	0.0	−0.5	0.0	$\{X_0, U_0\}$	$\{X_0, U_0\}$

Table 4.4: Summary of correlation structure for the predictors that were included in the models. The predictors were selected from X_0, U_0. The baseline outcome variable, Y_0 was forcibly included as a predictor in each model. The last two columns indicate the set of predictors retained for the best ANCOVA and change-score models, respectively, according to BIC

In the second analysis, Y_0 is available for selection together with U_0 and X_0 as candidate predictors for the follow-up, Y_1 outcome or change-score outcome $\Delta Y = Y_1 - Y_0$ outcome. A summary of the results for the scenario where Y_0 is not forcibly included as a predictor for the change score or follow-up outcomes in the models is shown in Table 4.5. The first five columns provides a summary for



(a)



(b)

Figure 4.1: (a) Predictors for the outcome Y_1 with Y_0 included by default (b) Predictors for the outcome ΔY with Y_0 included by default

the correlations between variables, as before. The last two columns are again the retained predictors for the two models based on different solution spaces where the implied covariance matrix is positive definite. From the results, we see that when Y_0 is not forcibly included in the model, there are sometimes some differences between the selected predictors for either ΔY or Y_1 as outcomes for the scenarios evaluated. This is evident when we compare the final model for each scenario. Some models retained the same set of predictors while other models retained different sets of predictors.

			Choose Y_0				
$\rho_{Y_0Y_1}$	$\rho_{X_0Y_1}$	$\rho_{U_0Y_1}$	$\rho_{U_0Y_0}$	$\rho_{X_0Y_0}$	$\rho_{U_0X_0}$	pred Y_1	pred ΔY
0.05	−0.70	−0.5	−0.5	−0.5	−0.5	$\{X_0, U_0\}$	$\{X_0, U_0, Y_0\}$
0.05	−0.95	0.5	0.0	−0.5	−0.5	$\{X_0, U_0\}$	$\{X_0, U_0, Y_0\}$
0.05	−0.70	−0.5	0.5	−0.5	−0.5	$\{X_0, U_0\}$	$\{X_0, U_0, Y_0\}$
0.5	0.10	0.5	0.0	−0.5	0.0	$\{U_0\}$	$\{X_0, U_0, Y_0\}$
0.5	0.20	−0.5	0.0	0.5	−0.5	$\{X_0\}$	$\{X_0, Y_0\}$
0.05	−0.95	−0.5	0.0	−0.5	−0.5	$\{X_0, U_0, Y_0\}$	$\{X_0, U_0, Y_0\}$
0.95	0.95	−0.5	0.5	−0.5	0.5	$\{X_0, U_0, Y_0\}$	$\{X_0, U_0, Y_0\}$

Table 4.5: Summary of correlation structure for the predictors that were included in the models. The predictors were selected from X_0, U_0 and Y_0. The last two columns are the set of predictors retained for the best ANCOVA and change-score models, respectively, according to BIC.

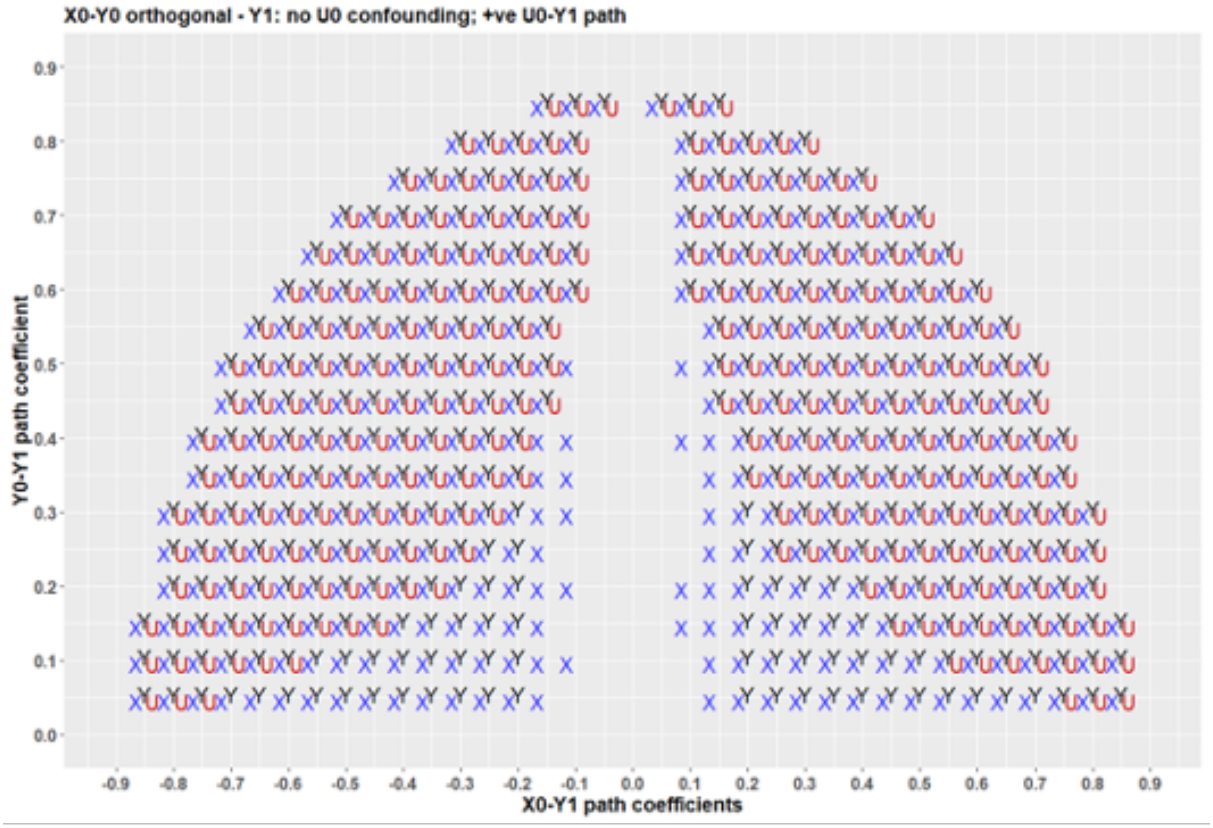


Figure 4.2: Predictors for the outcome Y_1 selected from X_0, U_0 and Y_0.

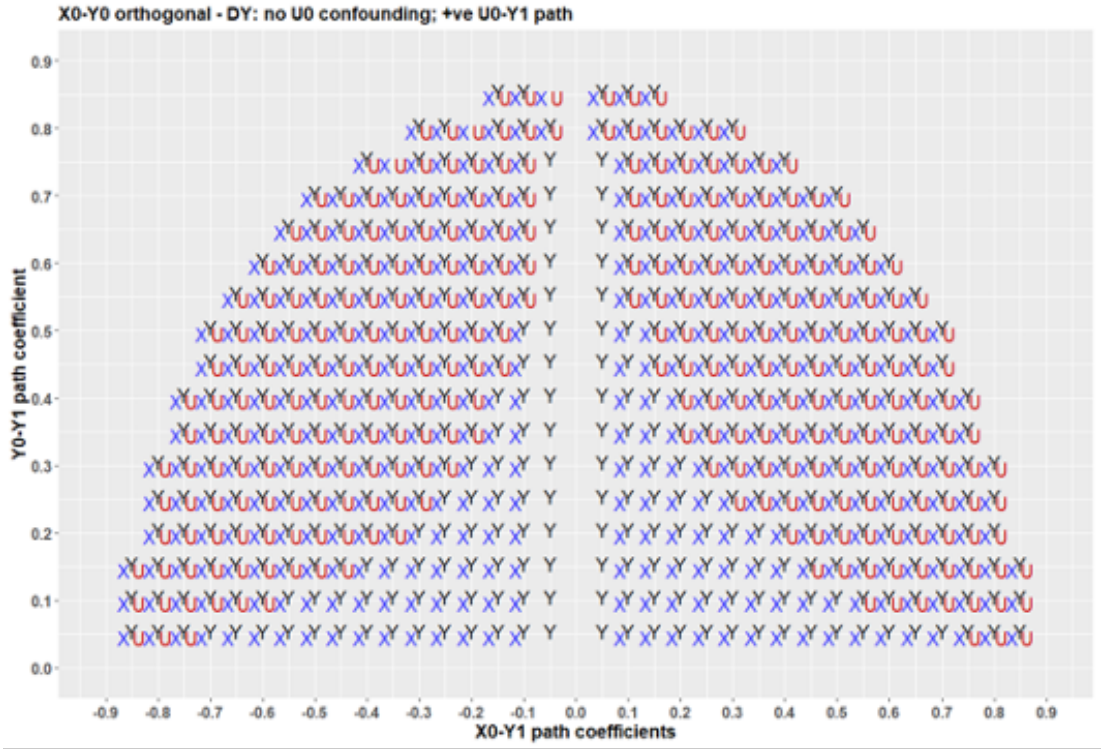


Figure 4.3: Predictors for the outcome ΔY selected from X_0, U_0 and Y_0.

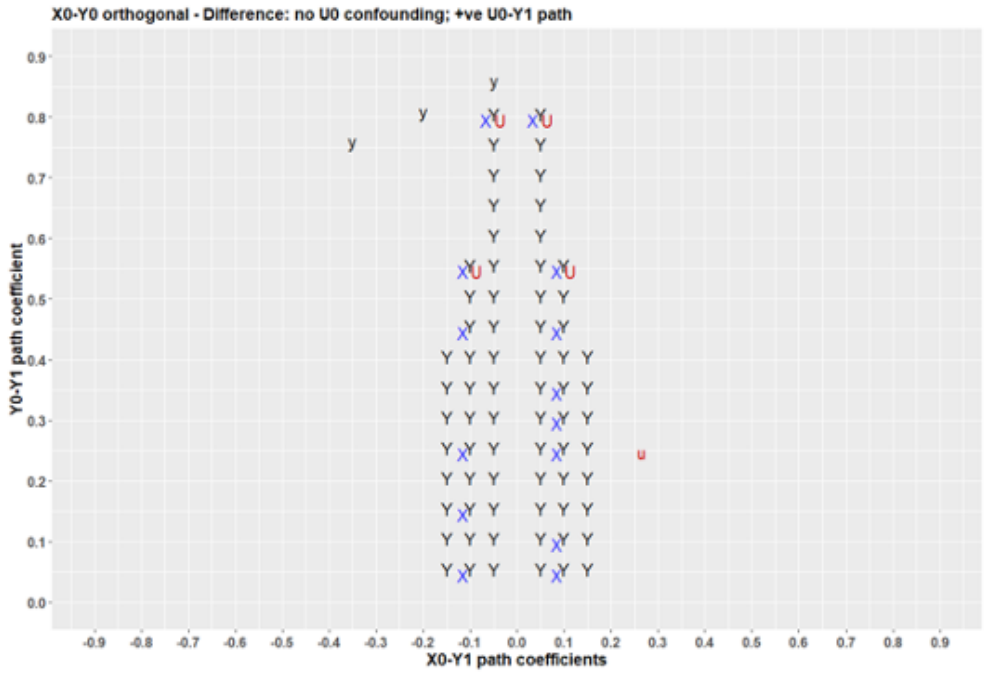


Figure 4.4: A graph showing the difference between the solution spaces for predictors selected in a model with Y as the outcome vs another model with ΔY as the outcome as shown in Figure 4.2 and Figure 4.3

The graph in Figure 4.2 shows the predictors selected for the outcome Y_1 when Y_0 is not forcibly included as a predictor. There are more marked patterns for either X_0 or U_0 than Y_0 suggesting X_0 and U_0 are mostly favoured as predictors for the Y_1 outcome model depending upon the implied covariance structure amongst X_0, U_0 and Y_0. X_0 is picked as a predictor if the correlation $X_0 - Y_1$ is close to 0 because of the indirect effect through Y_0. The graph in Figure 4.3 shows the predictors selected for the outcome ΔY when Y_0 is not forcibly included as a predictor in an ANCOVA model. The graph shows that Y_0 is mostly selected as a predictor of Y_1. There are marked patterns for the three predictors (i.e. X_0 , U_0 and Y_0) suggesting that all predictors are favoured as a predictors for the ΔY outcome model depending upon the implied covariance structure amongst

X_0, U_0 and Y_0. This pattern is consistent in all scenarios evaluated. The graph in Figure 4.4 shows the difference between the two graphs in Figures 4.2 and Figure 4.3. Upper-case letters are used depict selected predictors selected for the change-score model with an outcome, ΔY but not for the ANCOVA model with an outcome Y_1, while lower-case letters depict predictors that are selected for the outcome, Y_1 but not for outcome ΔY. There are some obvious differences between graphs in Figure 4.2 and Figure 4.3; it is clearly seen that the predictor Y_0 is more often not selected for the Y_1 outcome model than either X_0 or U_0 is not selected, and this is predominantly around circumstances where the path coefficient between X_0 and Y_1 is small, though occasionally patterned for much larger path coefficients between X_0 and Y_1, depending upon the implied covariance structure amongst X_0, U_0 and Y_0. Assuming the baseline outcome measurement, Y_0, were missing, we would be unable to consider it a candidate predictor and it would never be included as a selected predictor for either outcome model. We would then consider the two models with only X_0 and U_0 as candidate predictors. A summary of the results for this scenario where Y_0 is missing and hence not included as a predictor for the change-score or follow-up outcomes is shown in Table 4.6. The first five columns provides a summary of the correlations between variables, as before, while the last two columns are the root mean square error (RMSE) values for the two models evaluated on different solution spaces, where the implied covariance matrix is positive definite. From these results, we note that when Y_0 is not available for selection as a predictor in both models, the model with ΔY consistently performs poorly compared to the model with Y_1 as the outcome for the scenarios evaluated. This is evident when we compare the RMSE values for the two models. The RMSE values are higher in the model with

ΔY as the outcome suggesting that the model predictions are less accurate in the change-score model.

			Ignore Y_0				
$\rho_{Y_0Y_1}$	$\rho_{X_0Y_1}$	$\rho_{U_0Y_1}$	$\rho_{U_0Y_0}$	$\rho_{X_0Y_0}$	$\rho_{U_0X_0}$	Y_1_RMSE	ΔY_RMSE
0.05	−0.95	0.5	0.0	−0.5	−0.5	1.01	2.32
0.15	0.05	0.5	0.0	−0.5	0.5	1.33	1.43
0.15	0.10	-0.5	0.5	−0.5	0.0	1.29	1.42
0.2	−0.75	0.5	0.5	−0.5	0.5	1.18	1.04
0.95	0.95	0.5	−0.5	−0.5	−0.5	0.89	2.25

Table 4.6: Summary of the correlation structure for the predictors that were included in the models. The predictors were selected from X_0, U_0 and Y_0 is forcibly ignored in both models. The last two columns indicate the root mean square error values for the model with Y_1 and ΔY as outcomes

From the graphs in part (a) and part (b) of Figure 4.5, we see the different shapes of the plausible solution space (i.e. where the covariance matrix is positive definite). Examining the graph in part (c), we notice that patterns are seen for either X_0 or U_0 being favoured as a predictor for the Y_1 outcome model over the ΔY outcome model, or vice versa, depending upon the implied covariance structure amongst X_0, U_0 and Y_0. This shows considerable difference between the two model options, with potential sign reversal for some predictors in some instances.

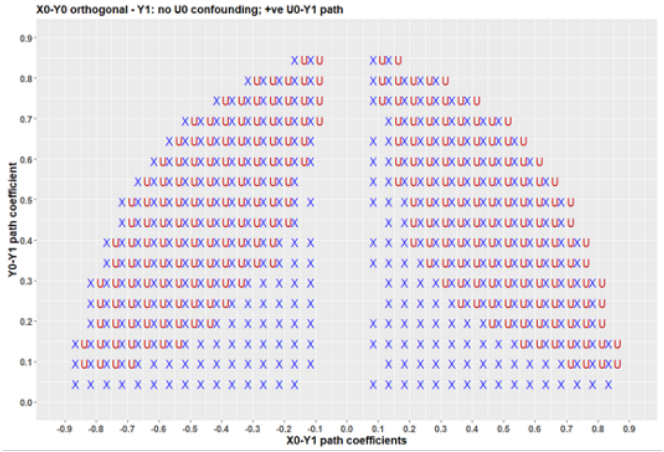
4.3.1 Conclusion

This chapter has demonstrated through simulations what we already knew, namely that provided Y_0 is conditioned for either a model (outcome Y_1 or the outcome ΔY), the two models are equivalent. It has also demonstrated what has not been examined before: (i) that allowing the two models to select Y_0 as a candidate predictor, different preferred models emerge from the training exercise; and (ii) that omitting Y_0 as a predictor in the model for Y_1 or ΔY results in a considerable differences in terms of prediction performance between the two models, with poorer performance predicting ΔY.

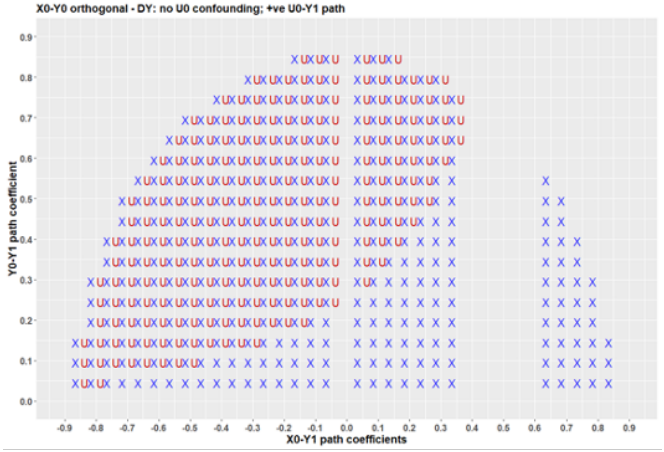
The first of these new observations raises serious questions around the validity of prediction models for change-scores, since it is clearly shown that the concept of change typically sought for causal inference is only obtainable by modelling follow-up outcome Y_1. Were this principle applied to prediction and the change-score calculated post-hoc, this could yield in some instances a different value to that derived through predicting ΔY directly. The question of which is the

'correct' change-score one should seek to predict might seem debatable for researchers not interested in causal inference, but causal knowledge underpins the data generating mechanisms evaluated and would therefore suggest that it is the change-score derived post-hoc that should be favoured.

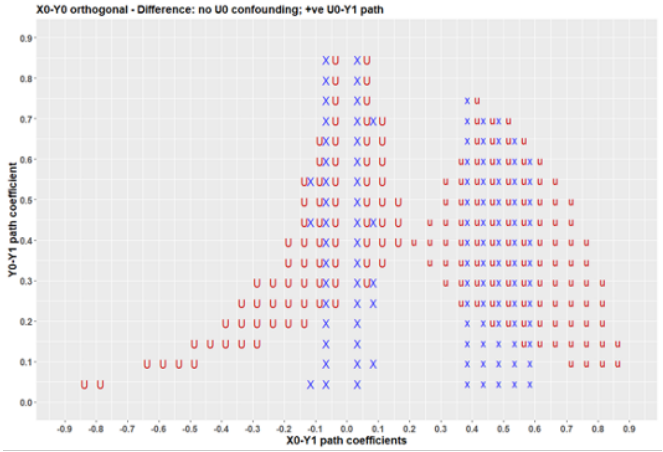
In any event, researchers need to be very cautious when predicting change in an observational study setting. The baseline, Y_0 should always be included where available as a predictor, whether predicting Y_1 or ΔY. Otherwise, failure to include the baseline outcome as a candidate predictor may yield incorrect predictions.



(a)



(b)



(c)

Figure 4.5: (a) Predictors for the outcome Y_1 selected from X_0 and U_0 with Y_0 forcibly excluded; (b) predictors for the outcome ΔY selected from X_0 and U_0 with Y_0 forcibly excluded; and (c) the difference between graph (a) and graph (b); upper-case letters are used depict selected predictors for the outcome, ΔY but not for outcome Y_1 , and lower-case letters depict selected predictors for outcome Y_1 but not for outcome ΔY

Chapter 5

Assessing the predictive acuity and clinical utility of survival prognostication amongst UK-HEART study patients using Statistical Modelling techniques

In this chapter we assess different statistical models for survival prognostication using the UK-Heart study dataset. We begin with the introduction which highlights some of the pitfalls of the commonly used statistical models and propose the extension of these models in a latent class framework.

5.1 Introduction

Risk prediction models (RPMs) remain popular for prognostication in cardiovascular medicine (DeFilippis et al., 2015; O'Donnell, 2020). While RPMs and their wider utility remain contentious beyond strict prognostication, and particularly in prevention (Arnold et al., 2020; Holmberg and Parascandola, 2010; Killu et al., 2019), many of the standard statistical modelling techniques commonly used are on clinical datasets that remain relatively small atleast when compared to contemporary notions of 'Big Data' (Diebold, 2012). A substantial statistical weakness of the commonest of these generalised linear models as a predictive tool is that they often fail to make full use of the joint information available amongst all candidate predictor variables. This is because these models rarely explore nonlinear relationships and interactions. Moreover, even when analysts optimally parameterise the candidate predictors available, and carefully consider all possible interaction terms between these, the clinical utility of GLMs is typically limited to predictions made at the population level (Arnold et al., 2020; Holmberg and Parascandola, 2010; Killu et al., 2019; Rockhill et al., 2000), while predictions at the individual level often lack precision. Although more sophisticated machine learning techniques may overcome the rigidity of GLMs and analysts' tendency to ignore nonlinear relationships and interactions, population-level predictions generated using cutting edge machine learning techniques will still be more reliable than individual-level predictions. Indeed, this bald fact applies to all prediction modelling techniques, including those underpinning contemporary claims of 'personalised' or 'precision medicine' (Wilkinson et al., 2020). It is therefore critical to recognise that while it is possible to determine what proportion of any given pop-

ulation will experience a specified outcome with a reasonable degree of accuracy, all such models provide less accuracy in determining outcomes for each individual within that population. Due to these caveats, predictions that are generated using GLMs cannot address the two key concerns of attending physicians:

- Which of the covariates are amenable to clinical intervention, so as to prevent any adverse outcome (or promote and amplify any favourable outcome) in each (or all) of these patients?
- Which particular patients will experience an adverse (or favourable) outcome?

To address the first of these questions, analysts need to switch their focus from predicting outcome values to estimating each of the relationships between covariates considered plausible targets for intervention and the outcome – an approach that can capitalise on recent advances in causal inference modelling techniques (Tennant et al., 2021b).

To address the second question, the best that can be achieved is to identify clinically meaningful subgroups of patients with shared characteristics that set them apart from other (subgroups of) patients using multivariable 'risk profiling'. Multivariable risk profiling can be achieved using latent class analysis (LCA) in which the exploration of nonlinearity, and of important interactions amongst included covariates, forms an integral part of classifying patients into subgroups (Dean and Raftery, 2010). Despite these benefits, the clinical utility of the resulting latent classes ultimately depends upon the extent to which this approach optimally exploits the joint information amongst available covariates. This approach perhaps has greatest clinical utility where there are: (i) factors known to be associated

with the outcome (which therefore facilitate prediction); but (ii) there are no known, modifiable causes of the outcome, or aetiological understanding is poor/-contested (as is the case with many rare, novel or complex diseases). Indeed, providing that the specified outcome is excluded from the LCA process (to avoid conditioning on the outcome) (Gadd et al., 2019), combining LCA class membership with candidate predictors provides increased complexity that can help exploit the joint covariate information in multivariable GLM prediction. That said, it is important to stress that causal interpretation of any covariate coefficients for latent class membership in such models remains deeply flawed for the very same reasons that causal interpretations of any covariate coefficient in prediction GLMs is flawed. Ostensibly this consideration might appear to limit the clinical utility of LCA-generated class membership, and it is true that describing class membership as a 'risk factor' often generates, and commonly reflects, a lack of understanding. Indeed, it risks conflating prediction and causal inference/determination just as it does when individual covariates are described in similar terms as 'risk factors' (Huitfeldt, 2016). Thus, while classifying subgroups of individuals using LCA can improve analytical practice and strengthen consideration of nonlinear relationships and important interactions amongst covariates, it does not address the clinical appetite for identifying so-called 'modifiable risk factors', or for individually tailored risk probabilities (the so-called 'holy grail' of personalised or precision medicine)(Rockhill et al., 2000). This might explain why the use of latent variable methods in prediction modelling remains largely under-explored, even though more sophisticated approaches exist that incorporate such techniques within GLM and offer substantial advantages for clinicians through subgroup risk profiling. These approaches involve the construction of la-

tent classes 'across' multivariable GLMs to: integrate consideration of nonlinear relationships and important interactions between covariates; and better capture (and exploit) the joint information amongst the available/included covariates. For example, in what is termed latent class regression (LCR) modelling, population data are partitioned into their constituent latent classes and a distinct GLM is simultaneously generated for each class. In the process, this approach accommodates any inherent population heterogeneity and thereby improves model precision.

5.1.1 Aims of this chapter

The aims of this chapter are:

1. To explore whether LCR models might improve the accuracy and precision of predictions at the population and individual level, by comparing LCR generated predictions to standard GLM.

2. To explore the use of LCA-generated class membership (Probabilistic or modal) variable as either the only candidate predictor in univariable GLMs, or as an additional candidate predictor alongside all other available covariates in multivariable GLMs might help to improve the predictive acuity in standard GLMs.

5.2 Data description

This chapter used data from the United Kingdom Heart Failure Evaluation and Assessment of Risk Trial 2 (UK-HEART2) a prospective cohort of ambulant pa-

tients with signs and symptoms of chronic heart failure (CHF) (Witte et al., 2018a). The study recruited 1,802 adult patients with CHF who attended specialist cardiology clinics in four UK hospitals between July 2006 and December 2014 (Witte et al., 2018b). Patients were eligible for recruitment if they: were aged 18 years or older; had had clinical signs and symptoms of CHF for at least 3 months; and had a left ventricular ejection fraction that was less than or equal to 45% (Witte et al., 2018a,b). Ethical approval was obtained from the research ethics committee at each participating hospital and eligible study participants were only recruited following informed consent (Cubbon et al., 2011). Additional information regarding UK-HEART-2's study design, patient eligibility and inclusion criteria, together with a detailed description of the study cohort has been reported elsewhere (Cubbon et al., 2011; Witte et al., 2018a,b).

5.3 Statistical methods

5.3.1 Variable selection and Model specification

The UK-HEART2 study data include 1802 patients with 88 variables. One of the challenging task was to select a subset of covariates to be used as predictors in the models. The following criteria was used:

1. Firstly, all potential predictors were checked for percentage missingness. All the predictors with more than 10% missingness (i.e. more than 180 missing values) were excluded from the analysis (38 covariates remained for further scrutiny)

2. Secondly, all the remaining candidate covariates were checked for prognostic

importance.

To simplify the methodological comparisons undertaken in the present study, only four covariates selected as candidate predictors comprising two demographic variables (age, sex), a single physiological parameter (haemoglobin level), and a single clinical characteristic (type 2 diabetes). These four covariates were then used to generate prognostic predictions of survival amongst UK-HEART2 participants using four separate statistical Procedures:

- Procedure 1 involved a single step multivariable Cox proportional hazard model that considered all four covariates as candidate predictors of survival, with no consideration of nonlinear relationships or interactions between covariates.

- Procedure 2 involved two sets of models, each involving two separate steps. Firstly, LCA was used to identify any latent classes or subgroups of participants using the four selected covariates, with individual membership to each latent class allocated using modal and probabilistic assignments.
  Secondly, a univariable Cox proportional hazard model examined latent class membership as the sole predictor of survival, with two separate models generated using latent class membership derived using modal (Procedure 2a) or probabilistic (Procedure 2b) assignment.

- Procedure 3 was an extension of Procedure 2 and it involved two sets of models, each involving two separate steps. Firstly, LCA was used to allocate latent class membership using modal (Procedure 3a) and probabilistic

(Procedure 3b) assignment.

Secondly, a multivariable Cox proportional hazard model considered all four covariates (as used in Procedure 1) plus latent class membership as multiple predictors of survival. Two separate models were generated using latent class membership derived using probabilistic assignment (Procedure 3a) or modal assignment (Procedure 3b).

- Procedure 4 involved single step latent class regression (LCR) models that considered all four covariates as candidate predictors to simultaneously predict both latent class membership and survival within each latent class.

5.3.2 Latent class model evaluation and classification diagnostic statistic

There are a number of model fit criteria that may be used for model evaluation in LCA to determine the final solution (i.e. a solution with an optimum number of latent classes). These are Bayesian information criteria (BIC), sample adjusted Bayesian information criteria (SABIC), Alkaike information criteria (AIC) as well as likelihood tests (Weller et al., 2020). In this analysis, covariate selection was guided by the desire to achieve parsimonious models according to the BIC, the statistic preferred as the most parsimonious penalised likelihood statistic to minimise the risk of overfitting (Hitchcock and Sober, 2004). In choosing the optimum number of latent classes for the latent variable models (i.e. LCA and LCR), BIC was again the preferred statistic as simulations have demonstrated it outperforms other model fit statistics (Nylund et al., 2007). Strategies for determining the optimal number of classes may also be influenced by interpretability

(such as clinical salience and/or utility (Gilthorpe et al., 2014; Harrison et al., 2013). In terms of model diagnostics in latent class models, entropy is reported which assesses the extent that individuals are aligned predominantly to a single class (i.e. it assesses how well a model is able to define latent classes), as this facilitates a clearer interpretation of each latent class as a near complete collection of individuals (Wang et al., 2017). Generally, an entropy value close to 1 is regarded as perfect and an entropy value above 0.8 is acceptable. The higher the entropy value, the lower the perceived misclassification error. It should be noted, however, there is no actual 'error' as such, since an entropy below 0.8 simply means a larger degree of uncertainty has been accommodated in the probabilistic classification process. A high entropy thus indicates that individuals are more aligned to a single class (large modal probability), which leads to clearer interpretation of each latent class (Celeux and Soromenho, 1996). A low entropy does not preclude latent classes having utility and substantive meaning, but individuals may not be as clearly aligned to just one class, making modal assignment a poor representation of the latent class structure. Model optimisation may thus depend upon both the overall predictive acuity of the latent class structure as evident from the model BIC and the intended utility of the determined classes thereafter as indicated by the model entropy.

5.3.3 Model selection and validation

All subsets regression was deployed (Kuk, 1984), along with k-fold cross-validation as recommended by Grimm et al. (Grimm et al., 2017), to find the best-fitting model for Procedures $1 - 4$, with four covariates considered for both Cox pro-

portional hazards models and (where applicable) the latent class models. The concordance statistic (c-statistic or c-index) was used to evaluate all models generated an approach that has been widely used in medical research to determine how well a risk prediction model could predict a higher risk score for a patient with an event than another randomly selected patient without an event (Hajian-Tilaki, 2013; Heagerty et al., 2000; Metz, 1978). In this way the c-index was used in this analysis to quantify the extent to which each modelling Procedure was able to assign a higher risk score to patients with shorter survival times and a lower risk score to patients with higher survival times. c-index values range from 0.5 to 1, where 0.5 indicates that the discrimination achieved is equivalent to (and no better than that that could be achieved) by chance; a value of 1 indicates perfect discrimination; and a value > 0.8 is interpreted as evidence of good discrimination. k-fold cross-validation involved randomly dividing the dataset into k partitions of approximately equal size, where $k - 1$ partitions were used as a training set and the model was evaluated and validated using the remaining kth partition, repeated k times. The value $k = 10$ was chosen based on established (and evaluated) best practice (Kuhn et al., 2013), with $k = 10$ favoured for less biased model parameters, according to experimentation (Harrison et al., 2013). The c-index was calculated for each of the 10 test samples, with subsequent confirmation of the results obtained from 10 iterations assessed using a bootstrap re-sampling procedure 100 times (creating datasets from the original data without making further assumptions) to provide empirical 95% confidence intervals (Bland and Altman, 2015).

5.4 Results

Table 5.1 provides a summary of the distribution of each covariate amongst participants in the UK-HEART-2 cohort. The mean age of the cohort's participants was 70 years, around two thirds (69.7%) were male and over a quarter (28%) had type 2 diabetes. The mean level of circulating haemoglobin was 13.5 g/dl; and 59% died during the period of follow-up (equivalent to a median survival of 3.4 years).

Table 5.1: Descriptive characteristics of the study cohort.

	Study Cohort N(%)
Participants	1,796 (100.0)
Deaths	1,061 (59.1)
Male	1,313 (73.1)
Type 2 Diabetes	504 (28.1)
	Median(IQR)
Survival Time (years)	3.40 (2.11, 5.78)
	Mean(95% CI)
Age (years)	69.7 (69.1, 70.2)
Haemoglobin (g/dl)	13.46 (13.38, 13.54)

N = number; % = percentage; IQR = interquartile range; CI = confidence interval.

Table 5.2: Latent class analysis (LCA) model summaries – the preferred model from this step was used in Procedures 2 and 3.

Latent Class Analysis model summaries						
Number of classes	**Number of parameters**	**BIC**	**Entropy**	**Class**	**Modal N (%)**	**Probabilistic N (%)**
1	6	19,818.53	-		1,796 (100.0)	-
2	11	19,537.79	0.75	Class 1	1,452 (80.8)	1425.3 (79.4)
				Class 2	344 (19.2)	370.7 (20.6)
3	16	19,445.74	0.74	Class 1	1,203 (67.0)	1175.0 (65.4)
				Class 2	480 (26.7)	500.7 (27.9)
				Class 3	113 (6.3)	120.3 (6.7)
4	21	19,422.35	0.80	Class 1	811 (45.2)	797.0 (44.4)
				Class 2	486 (27.1)	504.4 (28.1)
				Class 3	381 (21.2)	371.4 (20.7)
				Class 4	118 (6.6)	123.2 (6.9)
5	**26**	**19,421.44**	**0.67**	**Class 1**	**586 (32.6)**	**566.7 (31.6)**
				Class 2	**470 (26.2)**	**459.7 (25.6)**
				Class 3	**324 (18.0)**	**296.9 (16.5)**
				Class 4	**317 (17.7)**	**368.6 (20.5)**
				Class 5	**99 (5.5)**	**104.1 (5.8)**
6	31	19,422.87	0.63	Class 1	527 (29.3)	517.7 (28.8)
				Class 2	474 (26.4)	470.5 (26.2)
				Class 3	276 (15.4)	247.7 (13.8)
				Class 4	234 (13.0)	232.6 (13.0)
				Class 5	186 (10.4)	229.8 (12.8)
				Class 6	99 (5.5)	97.6 (5.4)

BIC = Bayesian information criterion; N = number; % = percentage; the optimal LCA model according to the BIC is emboldened.

Table 5.3: Latent class regression model with model fit statistics.

Latent Class regression model summaries						
Number of classes	**Number of parameters**	**BIC**	**Entropy**	**Class**	**Modal N (%)**	**Probabilistic N (%)**
1	3	3696.06	-		1,796 (100.0)	-
2	**10**	**3659.49**	**0.68**	**Class 1**	**1566 (87.2)**	**1425.3 (79.4)**
				Class 2	**230 (12.8)**	**370.7 (20.6)**
3	17	3682.45	0.91	Class 1	1064 (59.2)	1175.0 (65.4)
				Class 2	611 (34.1)	500.7 (27.9)
				Class 3	121 (6.7)	120.3 (6.7)
4	24	3728.31	0.61	Class 1	896 (49.9)	797.0 (44.4)
				Class 2	697 (38.8)	504.4 (28.1)
				Class 3	125 (7.0)	371.4 (20.7)
				Class 4	78 (4.3)	123.2 (6.9)
5	38	3822.50	0.94	Class 1	1064 (59.2)	566.7 (31.6)
				Class 2	606 (33.7)	459.7 (25.6)
				Class 3	120 (6.7)	296.9 (16.5)
				Class 4	4 (0.2)	368.6 (20.5)
				Class 5	2 (0.1)	104.1 (5.8)

BIC = Bayesian information criterion; N = number; % = percentage; the optimal LCA model according to the BIC is emboldened.

Table 5.4: Covariate coefficients for each preferred model (Procedures 1-4) executed on the complete data, along with median c-index and empirical 95% empirical confidence intervals generated through 10-fold cross-validation.

Model (c-index: 95% CI)		HR (95% CI)
Procedure 1 - CPH (c-index = 0.69: 0.67, 0.71)		
	Type 2 Diabetic vs. not	1.35 (1.16, 1.59)
	Male vs. Female	1.76 (1.47, 2.11)
	Age (per 5 years)	1.24 (1.20, 1.29)
	Haemoglobin (per g/dl)	0.82 (0.78, 0.86)
Procedure 2a - LCA (modal) / CPH (c-index= 0.65: 0.61, 0.67)		
†Class 1 (N = 586) vs:	Class 2 (470)	0.35 (0.30, 0.44)
	Class 3 (324)	1.33 (1.10, 1.60)
	Class 4 (317)	0.71 (0.57, 0.87)
	Class 5 (99)	0.17 (0.10, 0.29)
Procedure 2b - LCA (probabilistic) / CPH: (c-index = 0.65: 0.65, 0.66)		
‡Class 1 (32.0%) vs:	Class 2 (26.0%)	0.26 (0.19, 0.34)
	Class 3 (18.0%)	1.00 (0.71, 1.39)
	Class 4 (18.0%)	1.58 (1.27, 1.97)
	Class 5 (6.0%)	0.17 (0.09, 0.32)
Procedure 3a - LCA (modal) / CPH (c-index = 0.69: 0.66, 0.71)		
	Type 2 Diabetic vs. not	1.51 (1.13, 2.01)
	Male vs. Female	1.80 (1.49, 2.17)
	Age (per 5 years)	1.21 (1.13, 1.29)
	Haemoglobin (per g/dl)	0.82 (0.79, 0.86)
†Class 1 (N = 586) vs:	Class 2 (470)	0.77 (0.53, 1.10)
	Class 3 (324)	0.84 (0.59, 1.19)
	Class 4 (317)	0.92 (0.71, 1.20)
	Class 5 (99)	0.79 (0.38, 1.67)
Procedure 3b - LCA (probabilistic) / CPH (c-index = 0.69: 0.66, 0.71)		
	Type 2 Diabetic vs. not	1.44 (1.01, 2.06)
	Male vs. Female	1.70 (1.31, 2.21)
	Age (per 5 years)	1.21 (1.11, 1.32)
	Haemoglobin (per g/dl)	0.81 (0.76, 0.88)
‡Class 1 (32.0%) vs:	Class 2 (26.0%)	0.78 (0.41, 1.49)
	Class 3 (18.0%)	0.90 (0.55, 1.48)
	Class 4 (18.0%)	1.15 (0.56, 2.36)
	Class 5 (6.0%)	0.99 (0.35, 2.78)
Procedure 4 – LCR (c-index = 0.86: 0.84, 0.88)		
Cox proportional hazards model		
Class 1 ('High risk'):	Type 2 Diabetic vs. not	1.26 (0.91, 1.75)
	Male vs. Female	2.07 (1.58, 2.71)
	Age (per 5 years)	1.36 (1.28, 1.44)
Class 2 ('Low risk'):	Type 2 Diabetic vs. not	0.44 (0.23, 0.82)
	Male vs. Female	1.01 (0.64, 1.60)
	Age (per 5 years)	1.17 (1.06, 1.29)

c-index = concordance index; CI = empirical confidence interval obtained from the 2.5% to 97.5% centiles of bootstrapped samples following 10-fold cross-validation; HR = hazards ratio; OR = odds ratio; CPH = Cox proportional hazards; LCA = latent class analysis (modal assignment or probabilistic assignment); LCR = latent class regression.

Table 5.5: Summary of the odds ratios for the preferred Latent class regression model.

Class membership model		**OR (95% CI)**
'High' vs. 'Low' risk:	Type 2 Diabetic vs. not	0.27 (0.09, 0.76)
	Haemoglobin (per g/dl)	2.16 (1.64, 2.84)

Table 5.6: Descriptive characteristics for the 2-class Cox proportional hazards latent class regression model.

	Latent Class Regression Model			
	Class 1 ('High risk')		**Class 2 ('Low risk')**	
	Modal N (%)	**Probabilistic N (%)**	**Modal N (%)**	**Probabilistic N (%)**
Participants	1,566 (87.2)	1507.8 (84.0)	230 (22.8)	288.2 (16.0)
Deaths	1,046 (66.8)	1014.7 (67.3)	15 (6.5)	45.8 (15.9)
Male	1,160 (74.1)	1112.8 (73.8)	153 (66.5)	200.9 (69.7)
Type 2 Diabetes	368 (23.5)	342.3 (22.7)	136 (59.1)	162.5 (56.4)
	Median (IRQ)		**Median (IRQ)**	
Survival Time (years)	3.86 (2.41, 5.89)		1.13 (0.50, 2.27)	
	Mean (95% CI)		**Mean (95% CI)**	
Age (years)	69.2 (68.6, 69.9)		72.5 (71.1, 73.9)	
Haemoglobin (g/dl)	13.80 (13.72, 13.88)		11.14 (10.99, 11.30)	

N = number; % = percentage; IQR = interquartile range; CI = confidence interval.

Table 5.7: A summary of the performance for each model under 10-fold cross validation

Model	Median c-index	Minimum c-index	Maximum c-index
Standard Cox-PH model	0.69	0.61	0.72
LCR Cox model (Soft clustering)	0.86	0.68	0.91
LCR Cox model (Hard clustering)	0.77	0.63	0.84

Figure 5.1: A Scree-Plot showing fit-values for Latent Class Analysis Models

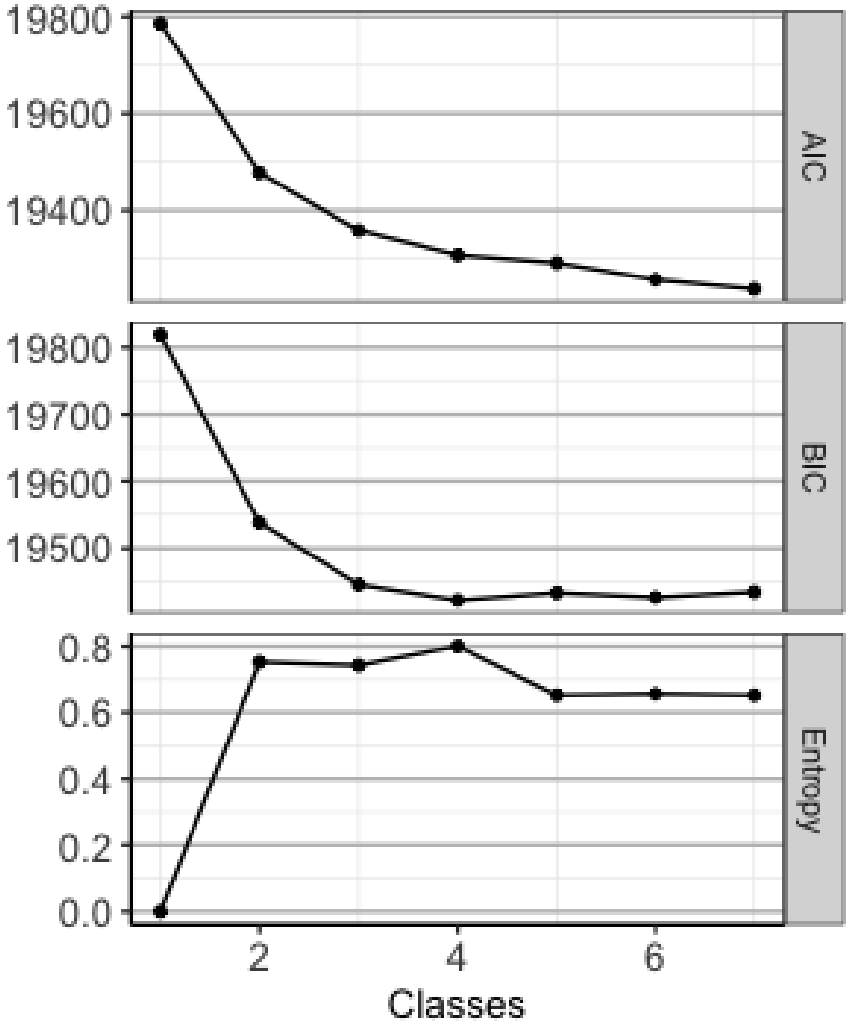
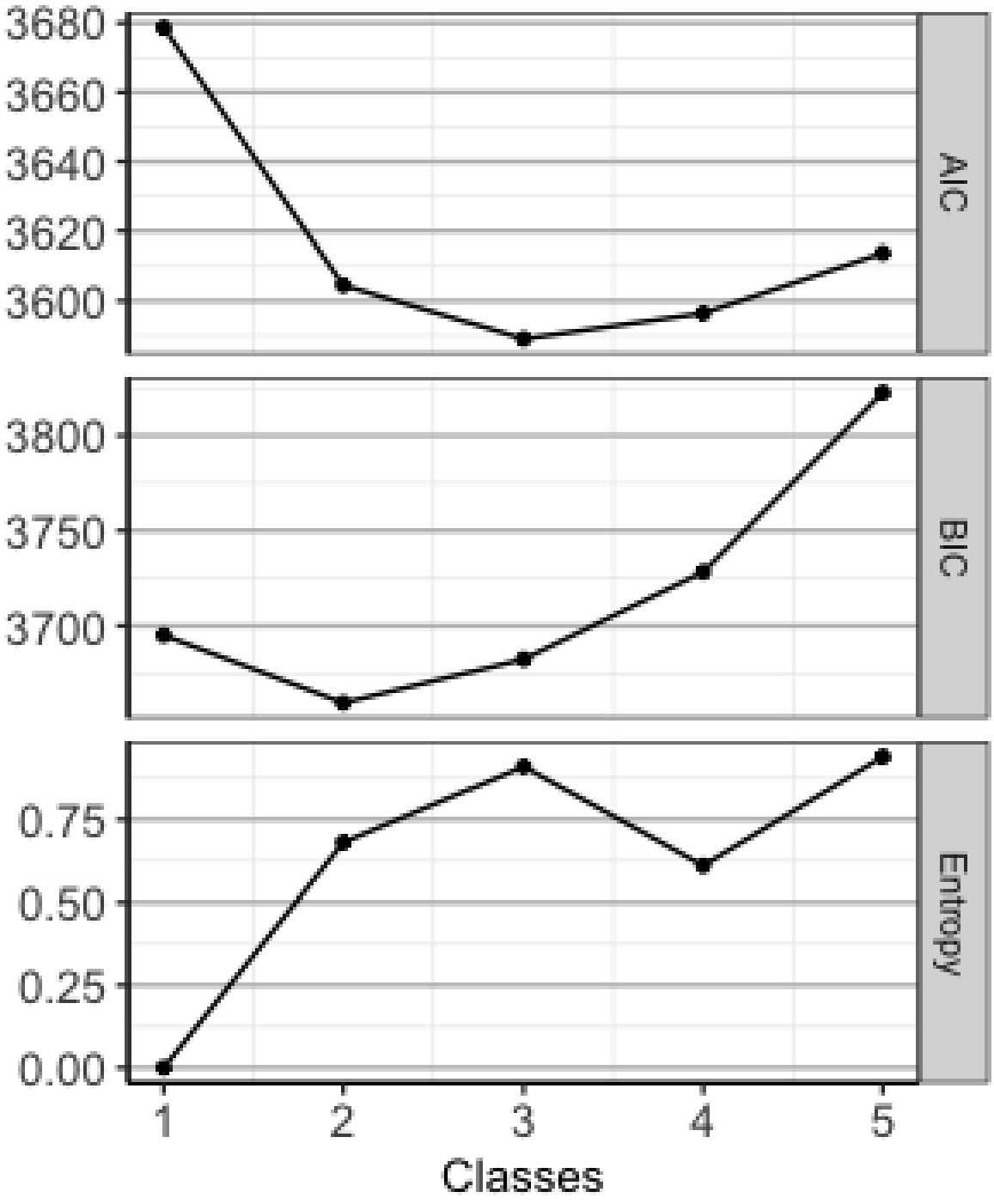


Figure 5.2: A Scree-Plot showing fit-values for Latent Class Regression Models.



In Procedure 1, the single step Cox proportional hazard model that considered all four covariates as candidate predictors of survival found that the model in which all four covariates were retained achieved the highest c-statistic (0.68) a level of acuity considered 'modest to poor' (Mandrekar, 2010). In Procedure 2, the LCAs conducted during the first step found that the 5-class model which retained all four covariates had the most favourable BIC (Table 5.2). Applying this 5-class model during the second step as the sole predictor of survival in a Cox proportional hazards model, achieved a c-statistic of 0.64 using modal assignment (Procedure 2a) and 0.65 using probabilistic assignment (Procedure 2b). These levels of acuity were both lower than that achieved using Procedure 1 (c-statistic=0.68). In Procedure 3, the second step involved consideration not only of the four covariates as candidate predictors of survival in the Cox proportional hazards model (as in Procedure 1), but also membership of the same 5-class model developed in the first step of Procedure 2. These analyses found that the best fitting model did not retain class membership as a predictor and forcibly retaining class membership in the model did not improve the c-statistics compared to what was achieved in Procedure 1, regardless of how class membership was assigned (modal: c-statistic = 0.68; probabilistic: c-statistic = 0.68). In Procedure 4, with all four covariates eligible for inclusion as candidate predictors of both latent class membership and the Cox Proportional Hazards models, some of the models were over-parameterised and failed to converge. Nonetheless, the most favourable of the models that successfully converged involved a latent class variable with just two classes (Table 5.3) and a c-statistic of 0.86 for models where soft clustering was deployed vs a c-statistic of 0.77 for models where hard clustering was deployed (Table 5.7). The improvement is down to the soft

clustering implicitly allowing for uncertainty in the latent class allocation and therefore provides a more robust model because it incorporates this latent class feature that addresses inherent heterogeneity amongst individuals. It is thus this very powerful feature that provides the step change in forming subgroups and prediction improvement. We also explored further by ignoring the uncertainty by considering each class separately. The majority class resulted in a lower c-index (0.57) compared to the minority class (0.79). It should however be noted that the hard classification deteriorates model robustness, as this neglects to exploit the very structural feature of latent decomposition that is utilised when we use the soft classification. It is this aspect of this approach which provides such improved predictions. When compared to the best performing models in Procedures $1-3$, these results suggest that Procedure 4 with soft clustering achieved a substantial improvement in predictive acuity of $18-22\%$.

Improvements in predictive acuity aside, the most favourable of the LCR models had only three of the covariates (age, sex, and type 2 diabetes) retained in the Cox proportional hazards models for each membership class, and only one of these covariates (type 2 diabetes) and the remaining covariate (haemoglobin level) retained as covariates in the LCR class membership model (Table 5.5). Sex and age were not covariates that contributed sufficiently to the class membership prediction model. As this sub model is a prediction model, focus is not on the coefficients of the model, rather the final predictions.

Given that all four covariates were retained in the most favourable CPH models generated by Procedures 1 and 3, and in the LCA models generated in the first step of Procedures 2 and 3, these findings suggest that Procedure 4's $18-22\%$ improvement in c-statistic is likely to have been achieved by exploiting the avail-

able covariate information differently to each of the three other Procedures. An indication of what this entailed can be found in the distribution of covariate characteristics amongst the two classes of the most favourable LCR model (Table 5.6), which suggest that these classes might warrant post-hoc labelling as '*high risk*' and '*low risk*' subgroups and might thereby offer substantial additional clinical utility in guiding the allocation of diagnostic, therapeutic, and/or palliative resources.

A further key finding that emerges from closer examination of the Cox proportional hazards models generated for each of the two classes within the optimum LCR model (Table 5.3) is that the contribution made by each of the covariates therein varied by class, and was dissimilar to the contribution these covariates made in those Procedures where all covariates were available for inclusion as separate candidate predictors (i.e. Procedure 1 and 3a/b). While the coefficient estimates of covariates in each of these models cannot be interpreted as measures of causal effects (Westreich and Greenland, 2013), their contribution as candidate predictors is strikingly different and depends upon the choice of model(s) used in each Procedure (Table 5.3). For example, the hazard of death associated with being male was 1.7 to 1.8 in Procedures 1 and 3, whilst for Procedure 4 being male was associated with a substantially higher hazard of death in one class (HR $= 2.07;\ 1.58, 2.71$) yet was unrelated to the hazard of death in the other class (HR $= 1.01; 0.64, 1.60$). Likewise, Type 2 diabetes was consistently associated with an elevated hazard of death in models generated under Procedure 1 and 3, while in Procedure 4 this covariate was associated with both an elevated hazard of death in one class (HR $= 1.26; 0.91, 1.75$) and a reduced hazard of death in the other class (HR $= 0.43; 0.23, 0.82$). Clearly, the joint information available

amongst each of the candidate predictors is selected and utilised very differently by each of the Procedures examined as seen in Table 5.4. Nonetheless, what sets the LCR model in Procedure 4 apart from the models used in Procedures $1-3$ is that LCR allows the predictive contribution from each covariate to be partitioned across any latent substructures existing within the study population, such that covariates are able to operate differently within each of the latent subgroups, thereby capturing and reflecting population heterogeneity that is unavailable to any of the other modelling Procedures; and, crucially, of substantial (additional) value when predicting the specified outcome.

5.5 Conclusions

The novelty in this chapter is that we have successfully compared three different approaches for incorporating latent variable methods within prediction modelling and demonstrated that LCR models with soft clustering can outperform not only the standard GLM (in which membership of latent classes is ignored – Procedure 1), but also those that include latent class membership identified using LCA to generate an alternative (Procedure 2) or additional candidate predictor (Procedure 3). This improvement in predictive acuity (which, as shown above, resulted in a $18-22\%$ improvement in c-index, despite the modest number of participants and covariates involved) illustrates the potential benefits of LCR for prediction modelling which, in this instance, shifted the acuity of prediction from 'modest to poor' to 'substantial' (Mandrekar, 2010).

We have further demonstrated how exhibiting the latent features through soft clustering that more explicitly addresses heterogeneity amongst individuals pro-

vides an improvement compared to the hard clustering approach in which heterogeneity is only partially exploited, and that the main benefit of latent structure is lost, showing some, but only modest, overall improvement.

We have further demonstrated that the latent class or subgroup structure that is revealed through LCR may have potential clinical utility. This is because it might as in the example examined here facilitate the identification of discrete subgroups (i.e. latent classes) of populations with very different underlying risks of the outcome. While such subgroups may not necessarily be amenable to effective intervention (given that LCR models support prediction, not causal inference (Westreich and Greenland, 2013)), they should help to improve the efficient allocation/targeting of outcome-relevant diagnostic, therapeutic and/or palliative resources to those subgroups identified as more likely to require (and perhaps even benefit from) these. However, to maximise the clinical exploitation of latent subgroups identified using LCR, model selection must focus on those achieving higher entropy where the probability of class assignment is closer to one for most assignments as this better aligns individuals/participants to a predominant single class (rather than aligning individuals/participants to multiple classes). For example, in Procedure 4, the 3-class model had lower predictive acuity but greater entropy than the 2-class model (see Table 5.1); and had the identification of clinically meaningful subgroups been the focus of these analyses, then it might have been appropriate to accept a modest reduction in predictive acuity in favour of enhanced clinical utility i.e. recognising three ('high', 'medium' and 'low' risk) subgroups rather than just the two ('high' and 'low' risk) subgroups identified by the LCR model with the most favourable predictive acuity (Table 5.2). Indeed, when clinical resources are scarce, such an approach might prove a more

reliable approach to resource allocation than one based upon the interpretation of predictive acuity alone.

Chapter 6

Evaluating the performance of Latent Class regression models using simulations that respect a causal process

In this Chapter, we discuss the second illustration of our simulation process described in Chapter 3. We begin by highlighting the potential benefits of electronic medical records and how these may be used to improve latent class regression modelling. This is followed by an illustrative example where we evaluate our proposed methods.

6.1 Introduction

The adoption of computerised medical records, digital medical devices, and data linkage protocols together with developments in high-powered computing have radically extended the potential scope, speed and accuracy of risk prediction models (RPMs). RPMs are derived by selecting predictors that are relevant to a particular circumstance and combining them into a multivariable model which can then be used to make predictions for the estimated risk or probability that a particular type of disease or condition is present in a person's body (or the probability that a disease or event will occur in future). There are many limitations of RPMs as discussed in Chapter 1, especially in the presence of heterogeneity which may affect the interpretation.

In contexts where population heterogeneity is prominent, any technique capable of embracing the cumulative consequences of variation should also provide novel insights into subgroup differences that substantively improve the accuracy of individual-level predictions. Latent class regression (LCR) is one such technique that may be used to aggregate data. It combines two distinct model concepts in a single estimation process and has better predictive acuity (and potentially greater clinical utility) than traditional regression-based generalised linear models (Mbotwa et al., 2021). LCR models allow both class membership and class-specific predictions of the target variable to make optimal use of the combined information contained in covariates (i.e. candidate predictors) such that each set of predictors may vary in relation to each specific analytical task and its related objectives (be that class membership or predictive acuity).

This flexibility in the choice of covariates used by LCR models in each of their

simultaneous analytical procedures has the potential to yield a step-change in parsimonious model complexity, and hence the accuracy of prediction, by exploiting more of the available covariate information to better predict the target outcome (directly and indirectly), leading to improved population and individual level predictions (Mbotwa et al., 2021).

With the recent developments in 'Big Data', new opportunities exist for exploring the latent class regression further by examining the covariate selection, parameterisation procedures by embracing the use of covariates (as candidate predictors) from a far broader range of sources and over much longer periods of time regardless of context or target variable.

6.2 Illustrative example

Assuming we are interested in an observational study setting where patients are followed up for a considerable number of years to study their survival from chronic heart failure (CHF) infections. To simulate data of this nature, the DGM described in Figure 3.13 was used. Simulated data were used to predict the survival of CHF patients, a context similar to the dataset used in a recent study (Mbotwa et al., 2021) which is discussed in Chapter 5. The candidate predictors were $\{X_1, X_2, X_3\}$, where each one represents a time-invariant phenomena that occur or time-variant characteristic that crystallize at discrete time points during an individual's lifecourse.

The first of these measured variables $\{X_1\}$ occurs in early life of an individual and therefore is the most distal to the target outcome (e.g death). Examples of predictors corresponding to $\{X_1\}$ might therefore include such features as geno-

type, size at birth, early life nutritional health or postnatal development. The second variable, $\{X_2\}$ is intermediate to the first and third variables, occurring midway through the lifecourse and might encompass such features as educational attainment, health-relevant lifestyles (e.g. obesity, lack of physical exercise and smoking) as well as occupation-related circumstances. The third of these, $\{X_3\}$ occurs in later life and is most proximal to the target outcome and might plausibly comprise recent symptoms and signs of CHF, adherence, or treatment behaviour post-diagnosis.

6.2.1 Aims

The aims of this chapter are:

1. To establish whether including covariates that are distal from the target variable (i.e. outcome variable) as candidate predictors of the heterogeneity may offer potential improvements in prediction acuity offered by latent class regression (LCR) models.

2. To assess the impact of dichotomising the candidate predictors to check if this may reduce the performance of LCR models.

3. To compare the performance and practical utility of standard 1-class Cox PH models and 2-class Cox PH LCR models in terms of their ability to generate accurate predictions of the target outcome given the simulated ground truth.

6.2.2 Data generating mechanisms

A data generating mechanism (DGM) can be described as a set of rules describing how data is generated. The data generating mechanism (DGM) adopted for the study used a temporal-causal framework as shown in Figure 3.13 in chapter 3 and simulated data for a range of plausible scenarios. The DGM assumes prior knowledge about the existing relationships between variables. Five multivariate normal variables, $X_1, X_2, X_3, \hat{S}, \hat{C}$ were simulated using ten path coefficients, $\rho_1, \ldots, \rho_{10}$. Each path coefficient or correlation represents the strength of a causal relationships amongst the five measured variables $\{X_1, X_2, X_3, \hat{C}, \hat{S}\}$, where three are candidate predictors $\{X_1, X_2, X_3\}$ and $\{\hat{S}\}$ is transformed into a measure of survival time S, plus $\{\hat{C}\}$ is also transformed into a latent variable, C, which is an assessment of population heterogeneity emerging from enigmatic variation during the lifecourse of individuals.

To simulate data for this experiment, eight scenarios were considered in which seven of the ten path coefficients $\{\rho_2, \rho_3, \rho_5, \rho_6, \rho_7, \rho_8, \rho_{10}\}$ were held constant, while three $\{\rho_1, \rho_4, \rho_9\}$ were assigned coefficients of either 0.5 to represent a strong causal contribution or relationship or 0.0 to indicate the absence of any causal influence or effect see Table 6.1 below.

Table 6.1: Summary of the eight scenarios for which data were simulated as the basis on which the performance and practical utility of standard 1-class Cox PH vs. 2-class Cox PH LCR models was evaluated in the present study together with a brief description of the distinct causal features within each of these scenarios.

	Path coefficients			
Scenario	$X_1 \rightarrow \hat{C}$	$X_1 \rightarrow \hat{S}$	$X_2 \rightarrow \hat{C}$	Distinct causal features
1	0.5	0.0	0.0	Only X_1 makes a strong contribution to $\hat{C}$; and X_1 has no direct causal effect on the target outcome $\hat{S}$.
2	0.5	0.5	0.0	Only X_1 makes a strong contribution to $\hat{C}$; and X_1 has a strong direct causal effect on the target outcome $\hat{S}$.
3	0.5	0.0	0.5	Both X_1 and X_2 make a strong contribution to $\hat{C}$; and X_1 has no direct causal effect on the target outcome $\hat{S}$.
4	0.5	0.5	0.5	Both X_1 and X_2 make a strong contribution to $\hat{C}$; and X_1 has a strong direct causal effect on the target outcome $\hat{S}$.
5	0.0	0.0	0.0	Neither X_1 nor X_2 makes any contribution to $\hat{C}$; and X_1 has no direct causal effect on the target outcome $\hat{S}$.
6	0.0	0.5	0.0	Neither X_1 nor X_2 makes any contribution to $\hat{C}$; and X_1 and has a strong direct causal effect on the target outcome $\hat{S}$.
7	0.0	0.0	0.5	Only X_2 makes a strong contribution to $\hat{C}$; and X_1 has no direct causal effect on the target outcome $\hat{S}$.
8	0.0	0.5	0.5	Only X_2 makes a strong contribution to $\hat{C}$ and X_1 has a strong direct causal effect on the target outcome $\hat{S}$.

These three key paths are through which distal and intermediate variables might plausibly affect the target outcome. This could happen either directly through the causal effects of phenomena occurring (or characteristics crystallising) in early life impacting directly on the risk of death from CHF in later life (e.g. $X_1 \rightarrow \hat{S}$) or indirectly through the causal contribution that distal and intermediate phenomena make to population heterogeneity during the lifecourse that mediates the (indirect) effects of these early and intermediate phenomena on risk of death from CHF in later life (e.g. $X_1 \rightarrow \hat{C} \rightarrow \hat{S}$ and $X_2 \rightarrow \hat{C} \rightarrow \hat{S}$) .

Of the seven path coefficients that were constant across the eight scenarios, the strongest (0.5) were assumed to be those between latent class membership, $\{\hat{C}\}$, the most proximal measured variable, X_3 , and the target outcome, $\hat{S}$. In contrast, those between the intermediate variable X_2 and the target outcome, $\hat{S}$; and between each successive measured variable (i.e. between X_1 and X_2; and between X_2 and X_3 were assumed to be weaker (0.3); while those between the most distal X_1 and most proximal X_3 measured variables, and between latent class membership $\{\hat{C}\}$ and the most proximal measured variable X_3, were both assumed to be weaker still (0.2). Each of the eight scenarios for which data were generated in the present study have been summarised in Table 6.1, together with a brief description of their distinct causal structures. In four of these (scenarios 2, 4, 6 and 8), the most distal measured variable X_1 is assumed to have a strong direct causal effect on the target outcome $\hat{S}$; whilst in the remainder (scenarios 1, 3, 5, and 7), the most distal measured variable X_1 has no direct causal effect on the target outcome $\hat{S}$. In two of each group of scenarios, the most distal measured variable X_1 makes a strong causal contribution to population heterogeneity (as captured by latent class membership $\{\hat{C}\}$, while in the remainder it makes

no such contribution. Finally, across each of these scenarios the intermediate measured variable X_2 makes either a strong or zero contribution to population heterogeneity $\{\hat{C}\}$.

6.2.3 Estimand

Our targeted estimand is the log hazard ratios. Patients with shorter survival times are expected to have higher log hazard ratios while those with longer survival are expected to have lower hazard ratios.

6.2.4 Methods

For each of the scenarios described above, continuous data were simulated for 200 datasets of 1000 cases per dataset. Each dataset comprised five multivariate normal variables $\{X_1, X_2, X_3, \hat{S}, \hat{C}\}$. The survival outcome, $\{\hat{S}\}$ was generated by exponentiating the normal variable, S to yield a right skewed outcome. We designed our simulations to have survival times ranging between 0 and 25 years. All exponentiated survival times above 25 were replaced by simulated data drawn from a uniform distribution with a minimum 0 and a maximum at 25. For simplification, censoring was zero across the hypothetical 25 year study. The binary latent class variable, $\hat{C}$ was derived by categorising the normal variable C at the 0.7 quantile to obtain two classes comprising 70% and 30% of the simulation sample, post-hoc labelled *low-risk* and *high-risk*, respectively. Finally, for the second set of simulations, the continuous predictors $\{X_1, X_2, X_3\}$ were also converted to binary variables by categorising each at the 0.7 quantile (all values below 0.7 were assigned zero while values above were assigned one).

6.2.5 Performance Measures

The performance and practical utility of standard 1-class Cox PH and 2-class Cox PH LCR models was formally assessed using 200 simulated datasets of continuous and dichotomised data for each of the eight scenarios. Separate models were generated using all three of the measured candidate predictors (distal, intermediate and proximal; $\{X_1, X_2, X_3\}$ as well as using two most recent of these predictors (intermediate and proximal $\{X_2, X_3\}$ for which measurements might be more readily available in most applied clinical settings. For each of the eight scenarios, two datasets (continuous vs. dichotomised); and three $\{X_1, X_2, X_3\}$ vs. two $\{X_2, X_3\}$ measured candidate predictors, the accuracy achieved by standard 1-class Cox PH and 2-class Cox PH LCR models was evaluated by the mean c-statistic and 95% simulation interval [SI]). The percentage of 2-class Cox PH LCR models that exceeded the accuracy achieved by standard 1-class Cox PH models was also assessed. Finally, the percentage of 2-class Cox PH LCR models that failed to converge was also recorded to provide a further indication of the practical utility of 2-class Cox PH LCR models vs. standard 1-class Cox PH models for the range of scenarios and datasets examined in the present study.

6.3 Summary of results

Tables 6.2 and 6.3 summarise the findings in terms of the correlation and covariance matrices of the data simulated under each of the eight scenarios for continuous and dichotomised predictors, respectively. The scenario-specific assessments of performance and practical utility of standard 1-class Cox PH against 2-class Cox PH LCR models and contrasts between models with all predictors $\{X_1, X_2, X_3\}$ against those using only the two most recent predictors $\{X_2, X_3\}$ are also presented. Model assessments were made in terms of mean (95% SI) c-statistic, percentage of 2-class Cox PH LCR models that failed to converge, and percentage of 2-class Cox PH LCR models whose accuracy exceeded that of standard 1-class Cox PH models.

The results confirm that 2-class Cox PH LCR models outperformed 1-class Cox PH models across all eight scenarios, regardless of the number of measured candidate predictors available (i.e. $\{X_1, X_2, X_3\}$ vs. $\{X_2, X_3\}$) or whether these were parameterised as continuous or dichotomised variables. These improvements were evident in terms of:

- Higher mean c-statistics (which averaged 0.72 and 0.67 amongst 2-class Cox PH LCR models using three and two candidate predictors, respectively; compared to averages of 0.69 and 0.64 amongst standard 1-class Cox PH models).

- The overall percentage of models that achieved higher c-statistics between standard 1-class Cox PH and 2-class Cox PH LCR models (which averaged 60.9% across all 32 pairs of models).

- The median percentage improvement in c-statistic between standard 1-class Cox PH and 2-class Cox PH LCR models (+6.0%; range: −11.3% to +21.1%) although this was even higher (+9.4%; range: -7.3% to +21.1%) with continuous predictor variables.

There was reduced performance of both sets of models when the number of measured candidate predictors available was reduced from three to two (i.e. from $\{X_1, X_2, X_3\}$ to $\{X_2, X_3\}$), and when information was lost through dichotomisation of continuous predictors. The 2-class Cox PH LCR models displayed a similar median decline in c-statistic (−6.7%; range: −15.9% to +4.2%) compared to the standard 1-class Cox PH models (−6.2%; range: −21.9% to +5.7%) when the number of predictors included was reduced from three to two, but a substantially greater median decline in c-statistic (−12.8%; range: −19.5% to −5.6% compared to −7.6%; range: −17.4% to 0.0% for standard 1-class Cox PH models) when the predictors were dichotomised. The potential improvements in 2-class Cox PH LCR models over standard 1-class Cox PH models was therefore more substantially diminished by loss of general predictor information than by the loss of distal predictor information. Indeed, in 9 out of 32 pairs (28.1% of models), 1-class Cox PH LCR models on average outperformed 2-class Cox PH LCR models in terms of c-statistic, this was twice as common when the predictors had been dichotomised and in most instances this occurred in scenarios where a relatively high percentage of 2-class Cox PH LCR models had failed to converge. These findings suggest that 2-class Cox PH models provide additional predictive value and greater practical utility in analytical contexts where distal predictors are unavailable or unmeasured, provided that predictors and/or their parameterisation offers more information.

Table 6.2: Covariance and correlation matrices derived for each of the eight scenario-specific datasets; together with model c-statistics and other summary measures, for standard 1-class Cox PH and 2-class Cox PH LCR models using all three $\{X_1, X_2, X_3\}$vs. only the two most recent $\{X_2, X_3\}$ candidate predictors as continuous variables to jointly predict survival together with C. Values are in red where the standard model on average outperforms the LCR model.

Path Coefficients			**D**					**M**		
			X_1	X_2	X_3	C	S	Model Predictors:	$X_1, X_2, X_3, \hat{C}$	$X_2, X_3, \hat{C}$
		X_1	1.00	0.30	0.39	0.16	2.46	Standard (1-class) Cox PH	0.63(0.55, 0.68)	0.60(0.51, 0.71)
$X_1 \to \hat{C}$	0.5	X_2	0.29	1.09	0.42	0.04	2.78	2-class Cox PH LCR	0.72(0.65, 0.82)	0.68(0.57, 0.72)
$X_1 \to \hat{S}$	0.0	X_3	0.34	0.35	1.28	0.12	4.25	LCR models failed	9%	3%
$X_2 \to \hat{C}$	0.0	C	0.35	0.09	0.24	0.21	1.31	LCR > Standard	98.9%	89.7%
		C	0.37	0.40	0.56	0.42	45.21			
			X_1	X_2	X_3	C	S			
		X_1	1.00	0.30	0.42	0.17	2.72	Standard (1-class) Cox PH	0.76(0.73, 0.80)	0.72(0.68, 0.76)
$X_1 \to \hat{C}$	0.5	X_2	0.29	1.09	0.53	0.19	4.07	2-class Cox PH LCR	0.78(0.67, 0.84)	0.76(0.58, 0.79)
$X_1 \to \hat{S}$	0.5	X_3	0.36	0.43	1.38	0.18	5.03	LCR models failed	8%	1%
$X_2 \to \hat{C}$	0.0	C	0.36	0.41	0.34	0.21	1.66	LCR > Standard	57.6%	90.9%
		C	0.39	0.55	0.61	0.52	49.59			
			X_1	X_2	X_3	C	S			
		X_1	1.00	0.30	0.39	0.15	4.23	Standard (1-class) Cox PH	0.73(0.65, 0.80)	0.57(0.50, 0.77)
$X_1 \to \hat{C}$	0.5	X_2	0.29	1.09	0.42	0.04	3.14	2-class Cox PH LCR	0.82(0.71, 0.86)	0.69(0.60, 0.73)
$X_1 \to S$	0.0	X_3	0.34	0.35	1.28	0.12	4.71	LCR models failed	6%	5%
$X_2 \to \hat{C}$	0.5	C	0.32	0.08	0.22	0.21	1.45	LCR > Standard	96.8%	87.4%
		C	0.59	0.42	0.58	0.44	50.51			
			X_1	X_2	X_3	C	S			
		X_1	1.00	0.30	0.42	0.17	4.02	Standard (1-class) Cox PH	0.83(0.81, 0.85)	0.81(0.77, 0.83)
$X_1 \to \hat{C}$	0.5	X_2	0.29	1.09	0.53	0.18	3.97	2-class Cox PH LCR	0.86(0.69, 0.89)	0.75(0.54, 0.78)
$X_1 \to S$	0.5	X_3	0.36	0.43	1.38	0.18	5.14	LCR models failed	2%	1.0%
$X_2 \to \hat{C}$	0.5	C	0.38	0.38	0.33	0.21	1.81	LCR > Standard	77.6%	2.0%
		C	0.58	0.54	0.63	0.56	48.91			
			X_1	X_2	X_3	C	S			
		X_1	1.00	0.30	0.29	-0.01	1.13	Standard (1-class) Cox PH	0.62(0.51, 0.72)	0.56(0.50, 0.64)
$X_1 \to \hat{C}$	0.0	X_2	0.29	1.09	0.039	0.00	2.55	2-class Cox PH LCR	0.66(0.58, 0.79)	0.66(0.57, 0.73)
$X_1 \to S$	0.0	X_3	0.26	0.34	1.21	0.06	4.15	LCR models failed	15%	0%
$X_2 \to \hat{C}$	0.0	C	-0.02	0.01	0.13	0.21	1.08	LCR > Standard	68.2%	95.0%
		C	0.16	0.35	0.54	0.34	49.12			
			X_1	X_2	X_3	C	S			
		X_1	1.00	0.30	0.32	0.05	1.48	Standard (1-class) Cox PH	0.74(0.69, 0.78)	0.62(0.55, 0.67)
$X_1 \to \hat{C}$	0.0	X_2	0.29	1.09	0.50	0.18	3.90	2-class Cox PH LCR	0.71(0.59, 0.78)	0.74(0.60, 0.78)
$X_1 \to S$	0.5	X_3	0.28	0.42	1.30	0.14	4.66	LCR models failed	10%	3%
$X_2 \to \hat{C}$	0.0	C	0.11	0.38	0.27	0.21	1.56	LCR > Standard	18.9%	96.9%
		C	0.21	0.53	0.58	0.48	50.03			
			X_1	X_2	X_3	C	S			
		X_1	1.00	0.30	0.29	-0.01	3.33	Standard (1-class) Cox PH	0.72(0.65, 0.77)	0.69(0.63, 0.75)
$X_1 \to \hat{C}$	0.0	X_2	0.29	1.09	0.39	-0.01	3.06	2-class Cox PH LCR	0.78(0.69, 0.84)	0.68(0.53, 0.74)
$X_1 \to S$	0.0	X_3	0.26	0.34	1.21	0.05	4.40	LCR models failed	9%	1%
$X_2 \to \hat{C}$	0.5	C	-0.01	-0.02	0.10	0.21	0.83	LCR > Standard	87.9%	34.3%
		C	0.47	0.41	0.56	0.26	50.39			
			X_1	X_2	X_3	C	S			
		X_1	1.00	0.30	0.32	0.05	3.39	Standard (1-class) Cox PH	0.70(0.59, 0.77)	0.74(0.66, 0.78)
$X_1 \to \hat{C}$	0.0	X_2	0.29	1.09	0.50	0.17	4.18	2-class Cox PH LCR	0.82(0.62, 0.86)	0.76(0.58, 0.79)
$X_1 \to S$	0.5	X_3	0.28	0.42	1.30	0.13	4.88	LCR models failed	4%	2%
$X_2 \to \hat{C}$	0.5	C	0.11	0.35	0.25	0.21	1.53	LCR > Standard	94.8%	63.3%
		C	0.47	0.56	0.59	0.46	51.67			

Table 6.3: Covariance and correlation matrices derived for each of the eight scenario-specific datasets; together with model c-statistics and other summary measures for standard (1-class) Cox PH and 2-class Cox PH LCR models using all three $\{X_1, X_2, X_3\}$ vs. only the two most recent $\{X_2, X_3\}$ candidate predictors as binary variables to jointly predict survival together with C. Values are in red where the standard model on average outperforms the LCR model.

Path Coefficients			**D**					**M**		
			X_1	X_2	X_3	C	S	Model Predictors:	X_1, X_2, X_3, C	X_2, X_3, C
		X_1	0.21	0.03	0.04	0.07	0.88	Standard (1-class) Cox PH	0.59(0.51, 0.68)	0.56(0.53, 0.66)
$X_1 \to C$	0.5	X_2	0.13	0.21	0.05	0.02	1.00	2-class Cox PH LCR	0.67(0.50, 0.76)	0.61(0.56, 0.69)
$X_1 \to S$	0.0	X_3	0.19	0.23	0.21	0.04	1.35	LCR models failed	39%	15%
$X_2 \to C$	0.0	C	0.32	0.10	0.20	0.21	1.31	LCR > Standard	67.2%	74.1%
		C	0.28	0.32	0.44	0.42	45.21			
			X_1	X_2	X_3	C	S			
		X_1	0.21	0.04	0.05	0.06	1.00	Standard (1-class) Cox PH	071(0.68, 0.74)	0.66(0.64, 0.69)
$X_1 \to C$	0.5	X_2	0.20	0.21	0.06	0.07	1.43	2-class Cox PH LCR	0.67(0.51, 0.76)	0.63(0.58, 0.68)
$X_1 \to S$	0.5	X_3	0.24	0.30	0.21	0.05	1.48	LCR models failed	47%	15%
$X_2 \to C$	0.0	C	0.30	0.33	0.24	0.21	1.66	LCR > Standard	22.6%	17.6%
		C	0.31	0.44	0.46	0.52	49.59			
			X_1	X_2	X_3	C	S			
		X_1	0.21	0.04	0.05	0.06	1.59	Standard (1-class) Cox PH	0.65(0.56, 0.73)	0.54(0.51, 0.69)
$X_1 \to C$	0.5	X_2	0.17	0.21	0.05	0.01	1.10	2-class Cox PH LCR	0.66(0.51, 0.77)	0.62(0.58, 0.69)
$X_1 \to S$	0.0	X_3	0.23	0.24	0.21	0.04	1.53	LCR models failed	41%	18%
$X_2 \to C$	0.5	C	0.26	0.06	0.18	0.21	1.45	LCR > Standard	57.6%	80.5%
		C	0.49	0.34	0.47	0.44	50.51			
			X_1	X_2	X_3	C	S			
		X_1	0.21	0.03	0.05	0.07	1.48	Standard (1-class) Cox PH	0.75(0.73, 0.76)	0.71(0.69, 0.72)
$X_1 \to C$	0.5	X_2	0.14	0.21	0.06	0.06	1.38	2-class Cox PH LCR	0.71(0.52, 0.79)	0.63(0.58, 0.68)
$X_1 \to S$	0.5	X_3	0.22	0.27	0.21	0.06	1.72	LCR models failed	49%	18%
$X_2 \to C$	0.5	C	0.34	0.30	0.27	0.21	1.81	LCR > Standard	15.7%	0.0%
		C	0.46	0.43	0.54	0.56	48.91			
			X_1	X_2	X_3	C	S			
		X_1	0.21	0.03	0.04	0.00	0.44	Standard (1-class) Cox PH	0.57(0.51, 0.67)	0.53(0.50, 0.62)
$X_1 \to C$	0.0	X_2	0.14	0.21	0.05	0.00	0.79	2-class Cox PH LCR	0.62(0.51, 0.79)	0.61(0.56, 0.69)
$X_1 \to S$	0.0	X_3	0.19	0.21	0.21	0.02	1.40	LCR models failed	40%	15%
$X_2 \to C$	0.0	C	-0.01	0.00	0.08	0.21	1.08	LCR > Standard	60.0%	90.6%
		C	0.14	0.25	0.44	0.34	49.12			
			X_1	X_2	X_3	C	S			
		X_1	0.21	0.04	0.03	0.02	0.54	Standard (1-class) Cox PH	0.69(0.56, 0.71)	0.62(0.52, 0.65)
$X_1 \to C$	0.0	X_2	0.17	0.21	0.05	0.07	1.43	2-class Cox PH LCR	0.67(0.51, 0.78)	0.62(0.58, 0.69)
$X_1 \to S$	0.5	X_3	0.14	0.23	0.21	0.04	1.43	LCR models failed	47%	18%
$X_2 \to C$	0.0	C	0.09	0.32	0.19	0.21	1.56	LCR > Standard	45.3%	57.3%
		C	0.17	0.44	0.44	0.48	50.03			
			X_1	X_2	X_3	C	S			
		X_1	0.21	0.04	0.04	-0.01	1.13	Standard (1-class) Cox PH	0.62(0.56, 0.72)	0.57(0.54, 0.68)
$X_1 \to C$	0.0	X_2	0.19	0.21	0.05	0.01	1.18	2-class Cox PH LCR	0.68(0.51, 0.76)	0.61(0.55, 0.69)
$X_1 \to S$	0.0	X_3	0.34	0.35	1.28	0.12	4.25	LCR models failed	34%	11%
$X_2 \to C$	0.5	C	-0.02	0.02	0.05	0.21	0.83	LCR > Standard	56.1%	62.9%
		C	0.35	0.36	0.46	0.26	50.39			
			X_1	X_2	X_3	C	S			
		X_1	0.21	0.04	0.03	0.02	1.15	Standard (1-class) Cox PH	0.65(0.53, 0.74)	0.68(0.54, 0.70)
$X_1 \to C$	0.0	X_2	0.19	0.21	0.06	0.06	1.50	2-class Cox PH LCR	0.68(0.51, 0.76)	0.63(0.55, 0.69)
$X_1 \to S$	0.5	X_3	0.16	0.28	0.21	0.05	1.56	LCR models failed	45%	14%
$X_2 \to C$	0.0	C	0.10	0.28	0.22	0.21	1.53	LCR > Standard	58.2%	22.1%
		C	0.35	0.46	0.47	0.46	51.67			

The results of the analyses presented in Tables 6.2 and 6.3 also reveal the extent to which the specific causal structures operating within each of the eight different scenarios appeared to affect not only the performance and practical utility of standard 1-class Cox PH and 2-class Cox PH LCR models but also the causal insights these models might provide.

When all three measured predictors were used, 2-class Cox PH LCR models were consistently better than standard 1-class Cox PH models in scenarios 1, 3, 5, 7, and 8; and consistently worst in scenario 6, regardless of predictor parameterisation. Likewise, when only the two most recent measured candidate predictors were used, 2-class Cox PH LCR were consistently better than standard 1-class Cox PH models in scenarios 1, 3, 5 and 6, and consistently worse in scenario 4. In other words, the three scenarios in which 2-class Cox PH LCR models consistently outperformed standard 1-class Cox PH models (regardless of the number of predictors or their parameterisation) were scenarios 1, 3 and 5; there was no scenario in which the latter consistently out-performed the former. This is striking since the common factor across these three scenarios (1, 3 and 5) is the absence of a direct causal path between the most distal measured variable X_1 and the target outcome S, such that the former only contributes to the prediction of the latter indirectly (i.e. mediated through its causal relationships with other predictors X_2, X_3 and latent class C. If the specific causal structure operating within such contexts affects the performance of different modelling strategies, then it is plausible that differences in the performance of such models might offer insight into the causal structure of underlying data generating mechanisms in real world contexts where the presence or role of enigmatic variation is unknown or uncertain.

Scenarios 1, 3 and 5 are not however the only instances in which there was no direct causal relationship between the most distal measured variable X_1 and the target outcome S, and it is unclear why the 2-class Cox PH LCR models generated under scenario 7, for instance, did not consistently outperform standard 1-class Cox PH models. Further exploration of this apparent anomaly is warranted to establish whether this finding is simply due to the chance phenomena, or the specific influence of key features in the causal structure of scenario 7 which distinguish this from scenarios 1, 3 and 5.

The first of these possibilities appears plausible given that in only one of the four sets of models concerned did the 1-class Cox PH models outperform the 2-class Cox PH LCR models (this being the models involving a reduced number of predictors parametrised as continuous variables), and the differences involved were modest and somewhat equivocal (mean c-statistic of standard 1-class Cox PH models was 0.69 [95% SI: $0.63, 0.75$] compared to 0.68 [95% SI: $0.53, 0.74$] for the 2-class Cox PH LCR, while the percentage of standard 1-class Cox PH with c-statistics that were larger than the equivalent 2-class Cox PH models was 65.7%.

The second possibility is perhaps less plausible, i.e. there is something peculiar to scenario 7 which favours 1-class Cox PH over 2-class Cox PH LCR, but only for models involving a reduction in the number of continuous predictors, and not for those involving dichotomised and/or additional predictors). In scenario 7, only the intermediate candidate predictor X_2 makes a strong contribution to C. Although the distal candidate predictor X_1 makes no direct contribution to either C or S, its exclusion from the models involving only two candidate predictors X_2, X_3 means that any contribution that X_1 makes to either C or S will

be entirely mediated through the modest causal path coefficients operating between X_1 and the intermediate X_2; $\rho_4 = 0.3$ and proximal predictors X_3; $\rho_5 = 0.2$. What is unclear is why this reduction in the information available for prediction should prefer standard 1-class Cox PH models over 2-class Cox PH LCR models only when the predictors are parameterised as continuous variables and not when the information available is further reduced through dichotomisation of predictors (where the mean c-statistic of the standard 1-class Cox PH = 0.57 [95% SI: $0.54, 0.68$] was actually 6.6% lower than that for the 2-class Cox PH LCR = 0.61 [95% SI: $0.55, 0.69$], and the percentage of standard 1-class Cox PH that exceeded 2-class Cox PH LCR models was only 37.1%; see Table 6.3).

6.3.1 Conclusion

In this chapter, we have demonstrated that the 2-class Cox PH model can perform better than the 1-class Cox PH LCR model given the simulated ground truth in many instances. It has also been shown that the overall performance of the models reduced when the distal candidate predictor was excluded from the model and it diminished further after dichotomisation of all candidate predictors. Therefore, we conclude that the Latent class regression can exploit latent heterogeneity to strengthen prediction and generate causal insight.

Table 6.4: A comparison of the median percentage improvement (+) or deterioration (-) in c-statistics achieved by 2-class Cox PH LCR models vs. standard 1-class PH models and median percentage in c-statistics achieved by models involving 2 vs. 3 candidate predictors, disaggregated by the parameterisation of predictors as either continuous or dichotomous.

	2-class Cox PH LCR vs. standard (1-class) Cox PH				**2 predictors** $\{X_2, X_3\&C\}$ **vs. 3 predictors** $\{X_1, X_2, X_3\&C\}$			
Models (n)	32				32			
Median	+6.0%				-6.5%			
Minimum	-11.3%				-21.9%			
Maximum	+21.1%				+5.7%			
	3 predictors $\{X_1, X_2, X_3\&C\}$		**2 predictors** $\{X_2, X_3\&C\}$		**1-class Cox PH**		**2-class Cox PH LCR**	
Models (n)	16		16					
Median	**+5.5%**		+6.3		-6.2%		-6.7%	
Minimum	**-5.6%**		-11.3%		-21.9%		-15.9%	
Maximum	+14.3%		+21.1%		+5.7%		+4.2%	
	Continuous	**Dichotomised**	**Continuous**	**Dichotomised**	**Continuous**	**Dichotomised**	**Continuous**	**Dichotomised**
Models (n)	8	8	8	8	8	8	8	8
Median	+7.4%	+3.1%	+9.4%	+3.5%	-5.0%	-7.0%	-4.1%	-7.4%
Minimum	-4.1%	-5.6%	-7.3%	-11.3%	-21.9%	-16.9%	-15.9%	-11.3%
Maximum	+14.3%	+13.6%	+21.1%	+15.1%	+5.7%	+4.6%	+4.2%	-1.6%

Chapter 7

A Cox neural network (Cox-nnet) model for survival prediction

The Cox Proportional hazard model is the most popular choice for modelling time-to-event data. The Cox PH enables one to assess the relationship between the patient's survival and a number of explanatory variables through the hazard ratios of the patients. Despite being a reasonably straightforward and very easy to interpret, a Cox PH model has some limitations when dealing with a large and complex datasets. One of the assumptions in a Cox PH model is that the baseline hazard function is common to all individuals in a population and that the relationship between the outcome and predictors is linear especially when dealing with continuous predictors. This assumption is not always true because in many ideal scenarios the relationship between the outcome and the predictors is non-linear.

Different machine learning (ML) methods have been widely proposed as useful alternatives to traditional statistical methods especially when the relationship

between variables is non-linear (Khan et al., 2020). ML methods are capable of determining complex, nonlinear relationships within datasets that cannot be easily captured using standard regression methods (Kuhle et al., 2018). A neural network extension of the Cox proportional hazard model known as Cox-nnet has been recently proposed (Ching et al., 2018). The Cox-nnet is fundamentally designed for survival prediction using high throughput genetic data. The Cox-nnet is a two-layer artificial neural network that is designed to perform cox regression on the output layer. The model is trained to minimise the partial log likelihood function using back propagation. The Cox-nnet has three main uses namely:

1. To provide a neural network alternative to the standard Cox model, especially when modelling non-linear relationships.

2. To help in revealing useful biological information through features of the hidden layer. The hidden layer of the Cox-nnet is capable of revealing relevant genetic pathways through the heterogeneity amongst the nodes.

3. To help in dimension reduction through analysis of the hidden nodes features.

The chapter is organised as follows, we begin by summarising some previously published work on applications of ANN in modelling survival data. We then apply the Cox-nnet on UK-Heart study data to predict survival of patients with chronic heart failure. We compare our findings to a standard Cox proportional hazards model and latent class Cox PH Model described in detail in Chapter 5.

7.1 Review of Applications of Machine learning (ML) in Survival prediction

The Cox Proportional hazard model is the most popular choice for modelling time-to-event data (Jerez et al., 2005; Ohno-Machado, 2001). The Cox PH model is a semi-parametric multivariate regression model that enables one to assess the relationship between the patients' survival and a number of explanatory variables. Different ML methods have been widely proposed as useful alternatives to the standard Cox PH model. Unlike the Cox PH model, ML methods makes fewer assumptions about the data. For example, ML do not require the assumption about proportional hazards to hold. The proportional hazards assumption stipulates that the ratio of the hazards for any two individuals is constant over a given period of time. ML methods are also preferred when the task requires exploiting complex relationships and interactions between several explanatory variables and the outcome of interest. In a standard Cox PH model, the effect of the covariates on the outcome is assumed to be linear on a log risk scale. In this section, we briefly review and discuss some of the ML methods that are used as alternatives to the Cox PH model.

A number of papers exist in the literature comparing different ML methods against traditional statistical models (TSM). In a survival analysis context, many researchers have attempted to compare the predictive performance of the Cox PH model against the artificial neural networks (ANN) in predicting survival of patients in different scenarios. We searched in Web of Science database for articles that contain information about applications of ML in survival prediction. We were particularly interested in neural networks methods and how these methods

can be used as alternatives to standard Cox PH model. The search terms used were ((Cox proportional hazard model) AND (Artificial neural network) AND (prediction)). Below are some of the examples with neural network applications in survival context:

Faraggi and Simon (Faraggi and Simon, 1995) proposed a Cox Proportional Hazard modelling approach for censored survival data using a feed forward neural network. Two NN models were constructed. The first one had 4 input variables, one hidden layer with 2 nodes and an output node. The second NN model has 4 input nodes, one hidden layer with three nodes and an output layer. The coefficients and hazard ratios of the semi-parametric Cox proportional hazards model are estimated using maximum likelihood.

Another study by (Jerez et al., 2005) used an ANN model to predict breast cancer relapse as well as to identify important prognostic factors in breast cancer relapse after surgery. A three layer ANN was used with every node in the input layer corresponding to the prognostic factor plus one node for the bias term. The number of nodes in the hidden layer was determined by exploring different network architectures. Different sets of variables were also used to determine the correct set of variables for the input node. Apart from the identified prognostic factors that were identified, followup time was also used as an input variable. The output of the network represent the cumulative probability of relapse for each patient. The network was trained using back-propagation (i.e. the input data was fed-forward through the model parameters towards the loss function. An error was calculated and propagated back to adjust the initial weights or parameters until the error was minimised) with gradient descent. The predictive performance of the neural network was compared against the standard Cox PH model. The neural network

model performed better than the Cox model.

Another study compared Cox PH model against a neural network model that was designed to have a categorical output (i.e. death at the end of a given interval 1 year, 2 years, 3 years or more than 3 years) (Ohno-Machado, 1997). This neural network model was used as a prognostic tool for people living with HIV/AIDs. The network was trained by back-propagation. A Cox PH model was built based on a particular set of predictors that were identified through backward selection process. The same set of variables were also used as inputs in the neural network model. The input constitutes patient demographics, laboratory markers and other clinical variables. Area under the curve was used to evaluate the predictive performance of these models. The standard approach yielded a similar predictive performance as the neural network model.

A more recent neural network extension of the Cox proportional hazard model known as Cox-nnet use a similar approach to that of Faraggi and Simon but with different optimisation functions to maximise the partial likelihood function. Faragi and Simon used Newton-Raphson iterations to maximise the loss function while (Ching et al., 2018) used different optimisation options namely: standard gradient descent, momentum gradient descent and Nesterov accelerated gradient.

7.2 Application of Cox-nnet Model to UK-Heart study data

In this section, the application of Cox-nnet is described using UK-Heart study data. We provide a detailed analysis procedure and summarise the results. We

compare the results from Cox-nnet prediction to that of LCM described in Chapter 2. We begin by defining hyper-parameters that are used to train a Cox-nnet model. We explore different Cox-nnet structures for improved survival prognosis. These results are discussed further in the discussion and conclusion chapter of the thesis.

7.2.1 Application of the Cox neural network

Using the UK-Heart study data, we trained different Cox-nnet models using 80% of the sample comprising of 1436 patients with four input variables namely, age, sex, diabetes status and haemoglobin content. We first performed a 5-fold cross-validation to determine a regularisation parameter that optimises the the loss function. After generating a parameter that optimises a loss function, new Cox-nnet models were built based on the modified loss function that incorporates extra information to penalise the large weights to curb overfitting. Possible methods for regularisation include ridge, dropout and a combination of ridge and dropout. The Cox-nnet models were tested using the remaining 20% of the data. The evaluation was repeated 10 times and the c-statistic was calculated for each replication.

We explored different network structures with different number of nodes in the hidden layer. We used four covariates to generate an input with four nodes with each covariate representing a node in the input layer. The hidden layer was experimented by varying the number of nodes from 1 to 10 and in each case we assessed the predictive acuity of the model using the c-statistic. We further explored Cox-nnet structures with 2 hidden layers. We present four Cox-nnet architectures

followed by a summary of the model parameters generated during the training process. A dropout regularisation parameter was used to avoid overfitting. To determine a regularisation parameter, we used a 5-fold cross validation using a randomly chosen 80 % of the training dataset and evaluated the performance on the remaining 20 % optimisation dataset. After generating our regularisation parameter, we updated our cost function and assessed our predictions. We assessed the performance of each model for all the 10 repetitions. We used a concordance statistic also known as the c-statistic as a prediction performance index for each model as advocated by Harrel (Harrell et al., 1982; Harrell Jr et al., 1984).

Scenario 1

One hidden layer with one node

Here, we consider a Cox-nnet structure with a single node in the hidden layer. The network has four input variables x_1, x_2, x_3 and x_4 which correspond to sex, age, haemoglobin content and diabetes status respectively. Each input variable is connected to the node in the hidden layer. Each connection between an input and the hidden node has an associated weight, w_{ij} where i represents the input while j represents a hidden layer. Thus, w_{ij} is read as weight between input node i and hidden node j. Each node in the hidden layer has an associated bias term, b_0.

The total input at node j is the weighted sum of k nodes in the input layer using equation 2.13. To generate an output at node j, we apply a *tanh* activation function on the output at node j using equation 2.15.

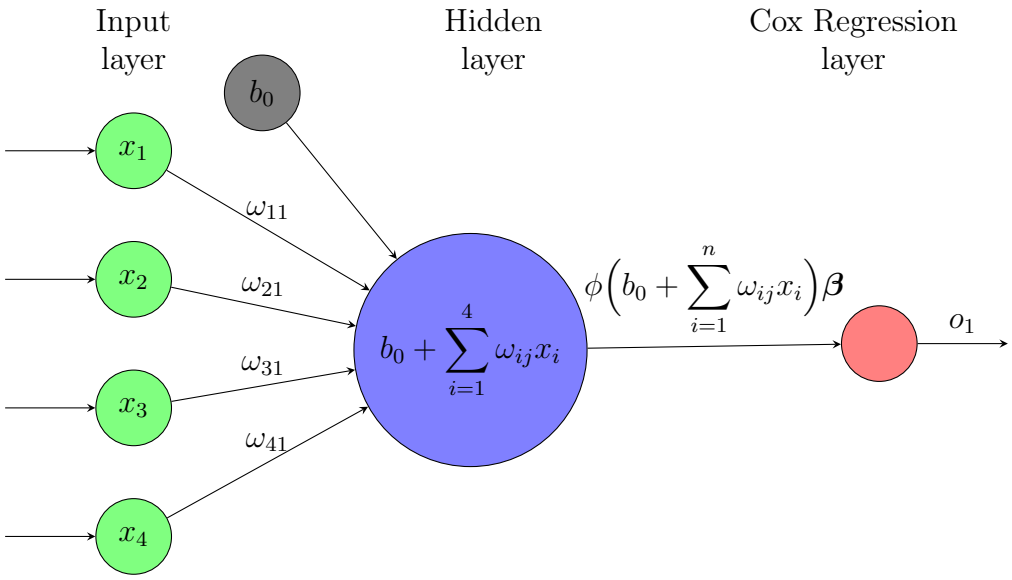


Figure 7.1: A single hidden layer Cox neural network with four input nodes and one hidden node and a single output node. x_0 is the bias term.

Scenario 2

One hidden layer with two nodes

In scenario 2, we consider a Cox-nnet structure with two nodes in the hidden layer. This network has four input variables x_1, x_2, x_3 and x_4 which have corresponding weights. The input at node j is found by multiplying each input by the weight and summing the product plus the bias term.

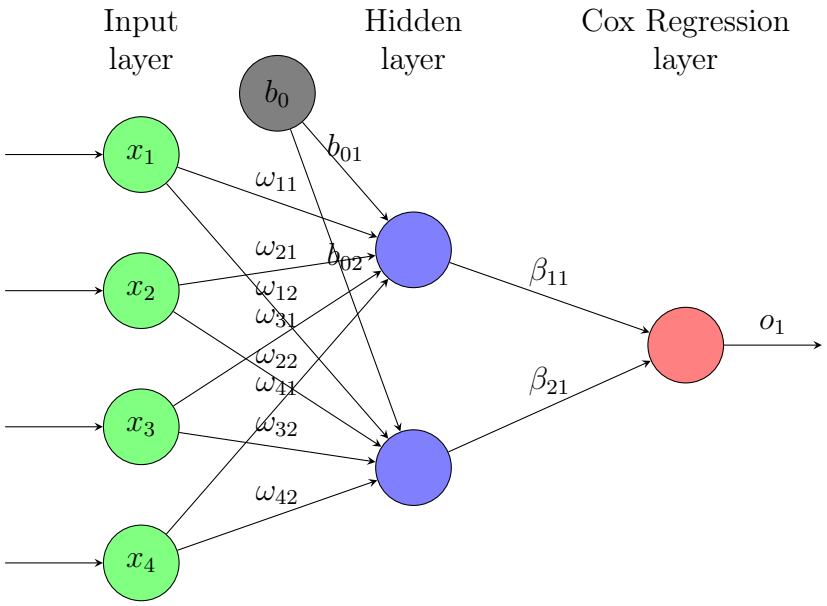


Figure 7.2: A single hidden layer Cox neural network with four input nodes, two hidden nodes and an output node. Each node in the hidden layer has a bias term b_0 .

Scenario 3

One hidden layer with ten nodes

In scenario 3, we consider a Cox-nnet structure with ten nodes in the hidden layer. This network has four input variables x_1, x_2, x_3 and x_4 which have corresponding weights. The input at node j is found by multiplying each input by the weight and summing the product plus the bias term.

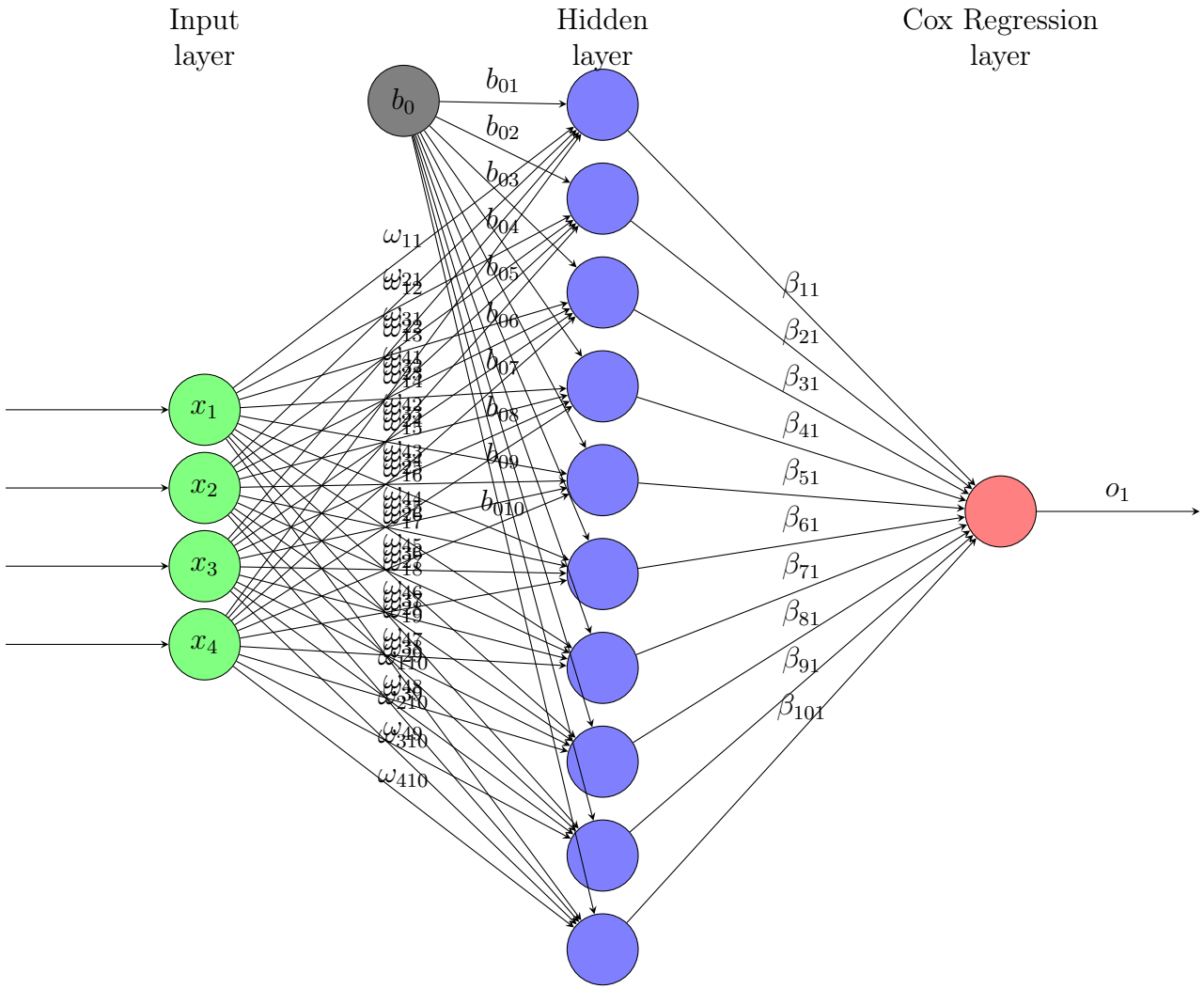


Figure 7.3: A single hidden layer Cox neural network with 4 input nodes and 10 hidden nodes and 1 output node.

Scenario 4

Two hidden layers with four nodes in the first layer and four nodes in the second layer

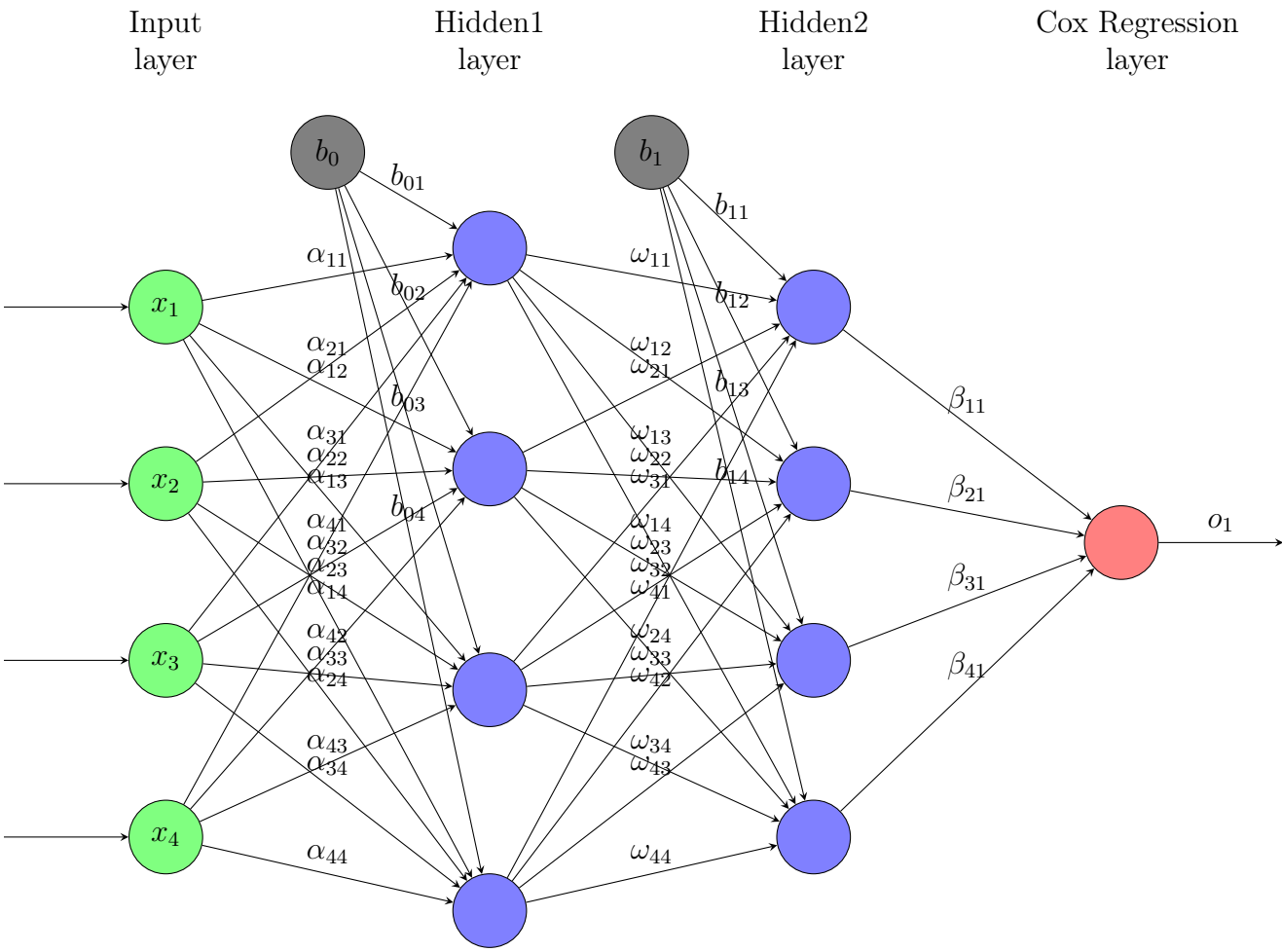


Figure 7.4: A single hidden layer Cox neural network with 4 input nodes and 4 hidden nodes and 1 output node. x_0 is the bias term.

7.3 Summary of results

7.3.1 Choice of an optimal regularisation parameter for the Cox-nnet Model

To identify an optimal regularisation parameter, 5-fold cross validation was used. A function, *L2CVProfile* was used to perform cross-validation on a list of values ranging from -4.5 to 0.5 returning a matrix of cross validated log likelihoods for each fold (i.e. 5 cross validated likelihoods were returned). To ensure that the model trains properly, a graph for the mean cross-validated likelihoods for each fold against the L2 parameter values was plotted to check if an optimal parameter was attained as shown in Figure 7.7 .

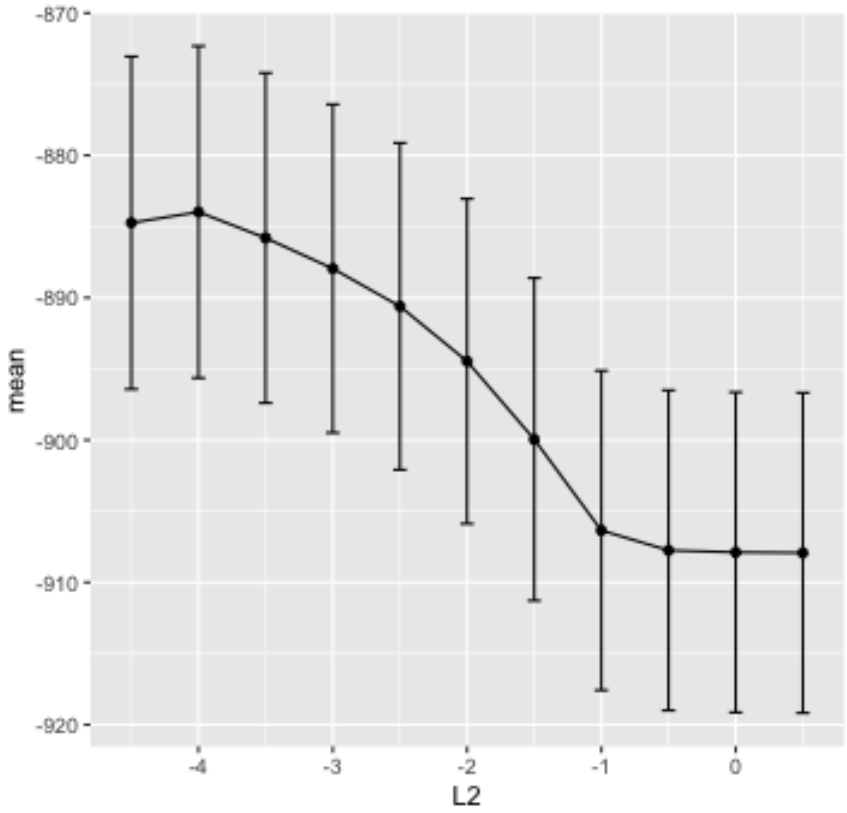


Figure 7.5: A Scree-Plot for the mean cross-validated likelihoods against the fitted L2 parameter values

A regularisation parameter (L2) was identified by choosing an L2 parameter value that optimised the partial likelihood function. A chosen parameter was added to the partial likelihood function. Cox-nnet structures with one hidden layer and two hidden layers were explored and compared. A two hidden layer Cox-nnet structure did not offer any improvement in terms of the c-index. In fact, a two layer Cox-nnet structure yielded a significantly lower c-index compared to the single layer network. As a result, a single hidden layer Cox-nnet model with ridge regularisation was chosen. A five-fold cross-validation was performed on the training set for a 1 to 10 nodes. A list of other parameters that were used

to train the Cox-nnet models include: Learning rate which was set to 0.001, the proportion of momentum was set to 0.99, The maximum number of iterations which was set to 2000. A random seed was set to 123 to ensure that the same subjects were selected for each fold. The stopping threshold was set to 0.995 to allow the training to stop if the cost function does not decrease by that proportion. The graph in Figure 7.6 shows the trend for the loss (i.e. the partial log-likelihood) vs. number of iterations for the training dataset. The partial log-likelihood begins by reducing significantly. As the number of iterations increases, we see that there is no further reduction in the cost function, suggesting that the model has fully trained and the optimum cost function has been attained. During the process of cross-validation nine folds were used to train the Cox-nnet model while the tenth fold was used to evaluate the performance. This process was repeated until a model was tested in each fold. The c-index was used to evaluate model performance in the test data. The summary of the results obtained from a 10-fold cross-validation are summarised in Table 7.1.

Table 7.1: Performance evaluation for cross-validated Cox-nnet models with different network architectures

Model	Mean (SD)	Median	Minimum	Maximum
Model 1	0.67(0.02)	0.68	0.65	0.69
Model 2	0.67 (0.03)	0.67	0.62	0.72
Model 3	0.67(0.02)	0.66	0.61	0.72
Model 4	0.66(0.03)	0.66	0.61	0.70

Model 1 is the Cox-nnet with 1 hidden layer with 1 node, Model 2 is the Cox-nnet with 1 hidden layer with 2 nodes, Model 3 is the Cox-nnet with 1 hidden layer with 10 nodes, Model 4 is the Cox-nnet with 2 hidden layer with 4 nodes in the first hidden layer and 2 nodes in the second hidden layer.

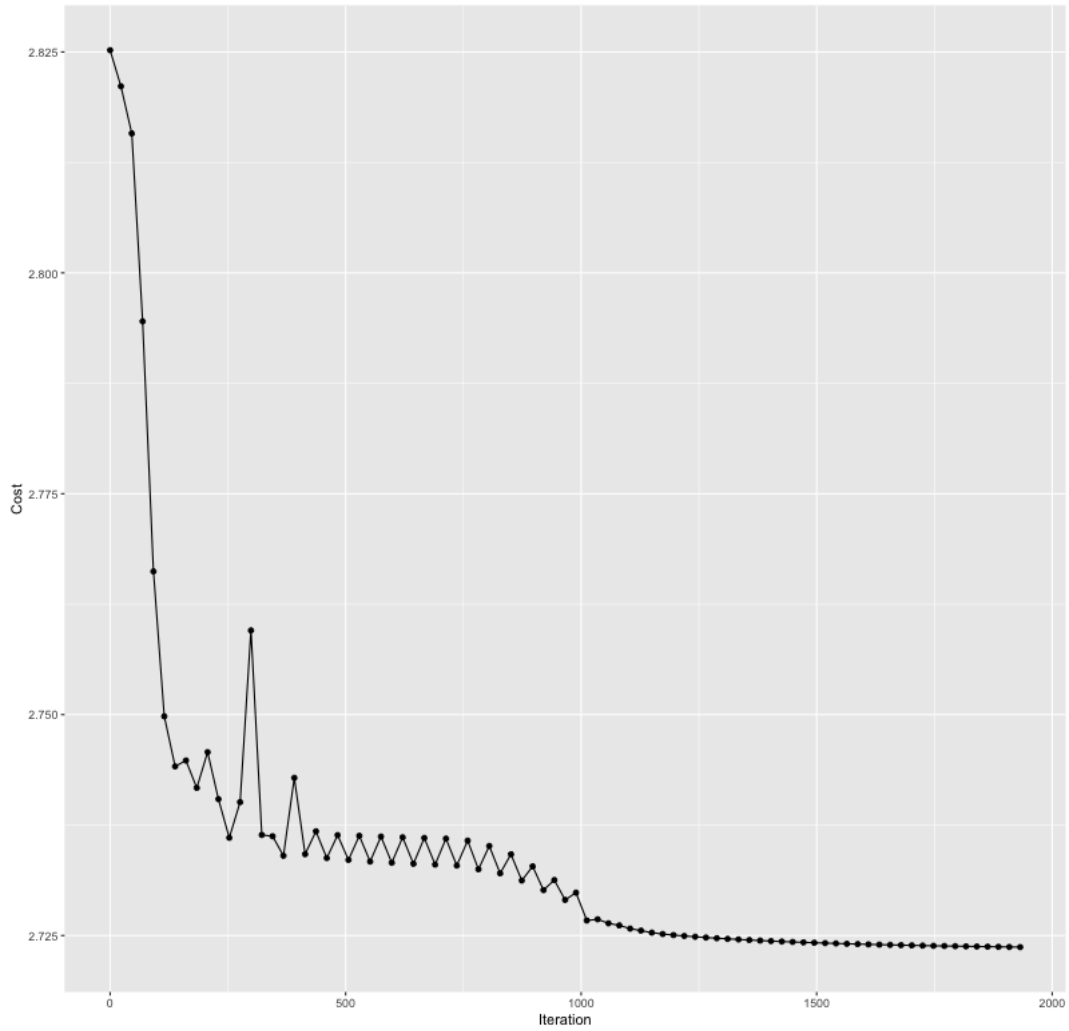


Figure 7.6: Graph showing Cost vs Iterations for a Cox neural network model with 2 nodes in the hidden layer

A *tanh* activation function was used to transform the output from the hidden layer to the Cox regression layer. We used the *predict* function to obtain the log hazard predictions (i.e. linear predictor) for the test dataset which are required when assessing the performance. To calculate the c-statistic we first created a data-frame with a vector of survival times and a censoring status indicator as well as all predictors used as input variables in the Cox-nnet model (age, haemoglobin content, diabetes status and sex). We then calculated the predicted log hazards for each patient based on the available predictors/ covariates . A c-index was calculated by first ranking the data according to the survival times. For every pair of patients that experienced the event, we looked at their survival and their log hazards. We then calculated the c-index as a proportion concordant pairs against the total (concordant and non-concordant). A pair was deemed concordant if a patient with a higher survival had a corresponding lower log hazard ratio.

7.3.2 Comparison with the standard Cox proportional Hazards Model and the Latent Class Cox regression model

Table 7.2 shows the overall performance for the standard Cox PH model, the cox neural network model and the latent class Cox regression model. A standard Cox PH model performed slightly better than the Cox-nnet model with a single hidden layer and two nodes (i.e. The median c-statistic for the standard Cox model was 0.69 while the Cox-nnet yielded a median c-statistic of 0.67.) The median c-statistic for the latent class Cox regression model was 0.86 which is higher than the median statistic for the standard Cox PH model (0.69) as well as the median c-statistic for the Cox-nnet model (0.67) which indicate that the latent class Cox

regression approach is able to discriminate better between high and low risk subgroups.

Figure 7.7 shows the box plots for the cross-validated c-statistics for three methods: Cox-nnet, latent class Cox regression model and the standard Cox regression model. It is clear that the latent class Cox regression approach outperformed the other methods. The latent class regression approach has an outstanding higher performance compared to other methods. This further confirms that the latent class Cox regression approach offers a better discriminatory ability compared to the standard Cox PH model and the Cox-nnet.

Table 7.2: A summary of performance for three models based on 10-fold cross validation

Model		c-statistic		
Description	Mean (SD)	Median (IQR)	Minimum	Maximum
Cox PH	0.68(0.04)	$0.67(0.65-0.70)$	0.62	0.72
Cox-nnet	0.67(0.03)	$0.69(0.67-0.71)$	0.61	0.72
Latent class Cox regression	0.83(0.08)	$0.86(0.84-0.88)$	0.68	0.91

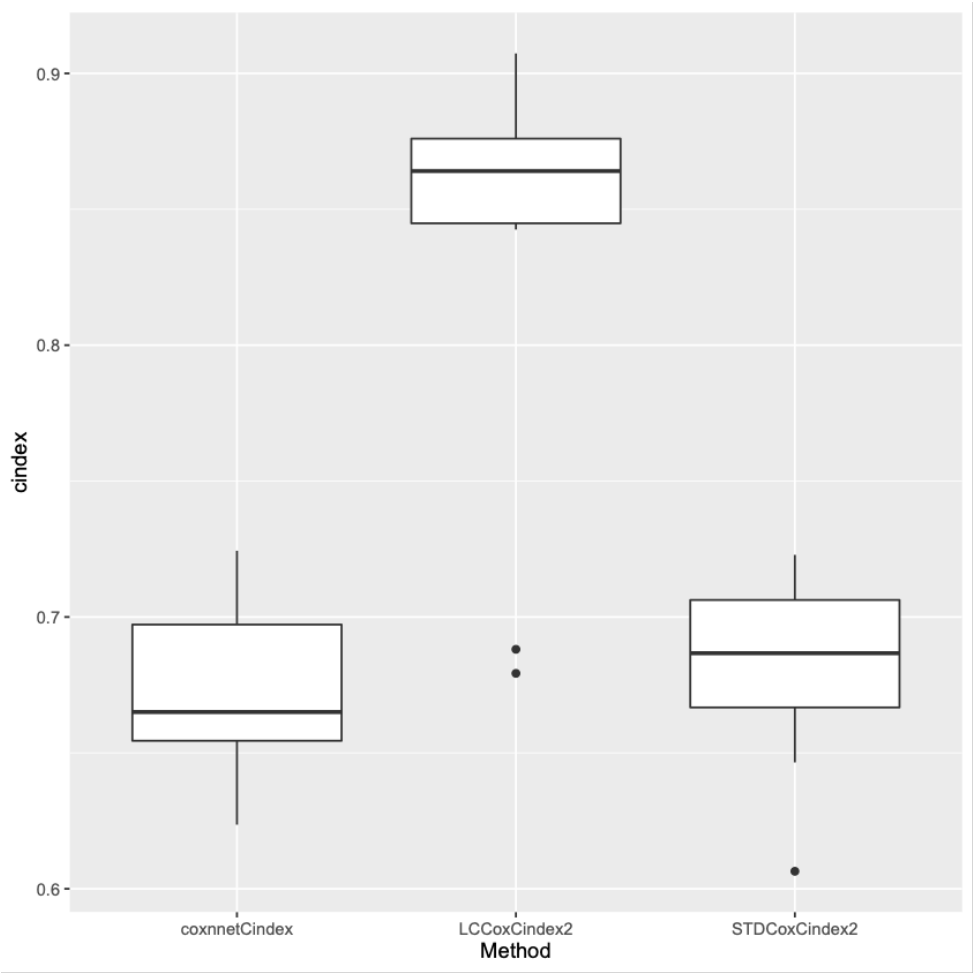


Figure 7.7: A boxplot for the distribution of the c-statistic for three models

7.4 Discussion

In this chapter, the aim was to compare the predictive performance of three prediction models using the concordance index (c-index). The c-index was used to compute the proportion of concordant pairs over all possible combinations. The c-index was constructed based on the prognostic index that was generated for each patient. A pair of observations was deemed to be concordant if a subject with highest prognostic index had a shorter survival time. The opposite was considered true for dis-concordant pairs. The first model is the standard Cox PH model with four predictors namely, age, sex, diabetes status and haemoglobin concentration. The second model is the Cox-nnet which is a neural network extension of the standard Cox PH model uses the same variables that were used as predictors in a standard Cox model as four input variables. The third model is the latent class Cox regression model which aimed at partitioning data into subgroups of latent classes and estimating separate risk models for each class. Latent class membership was predicted by diabetes status and haemoglobin content while separate Cox risk models used age, sex and diabetes status as predictors.

The results from a 10-fold cross validation show that the standard Cox PH model and the Cox-nnet exhibited a very similar performance in terms of c-index. However, the latent class Cox regression model performed better than the two methods. The LC Cox-PH model has worked better than the Cox-nnet because it accommodates the inherent population heterogeneity through modelled latent structure (i.e. through the soft clustering) which is either ignored in a standard Cox PH model or not fully captured when using the Cox-neural network. The Cox-nnet does not fully capture heterogeneity because its architecture is not ex-

plicitly designed to accommodate uncertainty like the Latent Cox Class regression model.

The standard Cox (i.e. Cox-nnet with no hidden layer) performed slightly better than the Cox-nnet with 1 hidden layer (2 nodes) [c-index: 0.69 vs 0.67]. This shows that the Cox-nnet with more parameters does not improve the quality of the model fit. A Cox-nnet with no hidden layer is therefore preferred because it has fewer parameters and therefore less prone to overfitting.

Comparing these methods in terms of their advantages, disadvantages and similarities we note that the Cox PH model is easy to implement and interpret the model findings. However, the proportional hazard assumption is not always true. The main disadvantage is that, unlike Cox-nnet, the interaction terms have to be manually added to the model when exploring non-linearity. This can be a challenging task especially when dealing with a large number of variables. The Cox-nnet does not require any prior assumptions e.g. the proportional hazard assumption. The Cox-nnet is also able to handle a large number of variables with ease, hence it is easier to investigate non-linearities and interactions. Cox-nnet uses a variable importance feature which calculates relative variable importance of a variable by drop-out method. This task is done by calculating the difference between the original log likelihood with all features and the new log likelihood without a particular feature. The difference is the variable importance score.

Cox-nnet's variable importance feature could help in choosing variables for inclusion in the standard Cox PH model and the latent class Cox model more especially when dealing with a large dataset where variable selection is usually problematic. The main problem is that Cox-nnet requires tuning a large number of hyper-parameters which is not an easy task and it may sometimes yield unre-

liable results due to overfitting.

On the other hand, an advantage of the Latent class Cox regression over Cox-nnet and Standard Cox PH model is that it relaxes an assumption of having a global risk model. It allows formation of risk subgroups and subsequently estimates separate risk models for each subgroup. Latent Class Cox regression uses a standard statistical diagnostic approach to determine the correct number of latent classes using Bayesian information criteria (BIC), but the Cox-nnet does not have any reliable diagnostic statistic to determine the correct architecture. As a result, Cox-nnet requires more time to experiment with a range of parameter combinations and assessments. A Latent Class Cox regression approach allows inclusion of covariates to explain class-membership thereby improving its clinical usefulness.

In conclusion, the aim of this chapter was to compare the predictive acuity of the standard Cox PH model, the Latent Class Cox regression model as well as the Cox-nnet model which is a machine learning version of the standard Cox PH in predicting survival of patients with Chronic heart failure using four predictors, age, sex, diabetes status and haemoglobin content. It has been illustrated that the Latent Class Cox regression model provides a superior approach to modelling the survival of patients with chronic heart failure as compared to the standard Cox PH and Cox-nnet models based on the c-statistic.

Chapter 8

Conclusions

In this chapter, we provide a summary of the main findings from this thesis followed by limitations of current work and recommendations for further research. This thesis has six objectives and chapter 3 -7 have addressed all the objectives.

8.1 Summary of main findings

In this thesis, we explored statistical and machine learning models to determine if they offer any improvement in predicting either change (in health status) or specific discrete events (e.g. death) within an observational study setting. The motivation behind this research has been to find ways of improving prediction of group and individual outcomes to facilitate the delivery of personalised care.

In Chapter 3 we addressed objective number 1 and we demonstrated how a DAG can be used to used to simulate data that respects a defined causal structure. We also highlighted the benefits of using a DAG to simulate data as opposed to a naive approach that uses the observed covariance structure. The main benefit

of adopting DAG based simulations as demonstrated in this chapter is that it offers a flexible way of exploring scenarios which may not be easily captured by simply specifying an observed covariance structure. Two examples were explored to demonstrate how complex scenarios are easily captured with a DAG based approach. The first application we explored is about a prediction of change problem while the second application is about predicting the survival or death amongst patients with chronic heart failure (CHF) in a heterogeneous population. The main achievement in this chapter is that we successfully integrated a causal thinking in our data generating process which is novel. The main message in this chapter is to emphasise the importance of understanding the data generation process when exploring prediction problems to avoid making incorrect predictions.

In Chapter 4 we addressed objective number 2. We explored the first application from Chapter 3 in which we aimed at evaluating the two methods (i.e. change-score and regressor method) that are commonly used to analyse change or followup. The aim in this chapter was to assess the implications of including or excluding the baseline measurement when predicting change or followup. The results from the simulations revealed that both the change score and regressor approach yield the same output provided the baseline measurement is conditioned for, in each model. It was also demonstrated that forcibly excluding the baseline measurement as a predictor in the change-score prediction model affects the precision of the estimates generated from the model as evidenced by the higher mean square errors compared to the regressor method. Again, this is the first time extensive simulations have been used to confirm that the change score and regressor methods yield the same output when the baseline measurement is conditioned for, in each method. Another novel finding is that when we forcibly

exclude the baseline measurement as a predictor in the change-score method, we get wrong predictions and the results may therefore be misleading.

In Chapter 5, we addressed objective number 3. We examined four different modelling procedures to assess whether LCR models may offer improved prediction and clinical utility over traditional regression methods using the real world demographic and clinical data from $1,802$ heart failure patients enrolled in the UK-HEART2 cohort. The LCR model demonstrated a substantial improvement in predictive acuity and clinical utility over traditional methods. The standard regression approaches which are frequently used fail to generate reliable predictions in the presence of heterogeneity which is usually overlooked and it leads into poor predictions and wrong inferences. LCR modelling resulted in an $18-22\%$ improvement in predictive acuity over alternative standard models which clearly showed that standard statistical models are limited because these methods do not take into account the inherent heterogeneity and therefore lack clinical utility. We therefore concluded that LCR modelling can improve the predictive acuity of GLMs and enhance the clinical utility of their predictions.

In Chapter 6 we addressed objective number 4 by exploring the second application of the simulations generated in Chapter 3. The aim of this chapter was to examine whether using distal, intermediate and proximal information about patients combined with causal reasoning might help to improve prediction when using latent class regression modelling. We also investigated the impact of dichotomising candidate predictors on the overall prediction. From the results, two class latent class Cox PH model showed an improvement in terms of predictive acuity over the standard 1-class Cox PH model. Therefore, we concluded that integrating a causal insight within the latent class regression modelling can exploit

latent heterogeneity to strengthen prediction.

Lastly, in Chapter 7, we addressed objective number 5. The aim of this chapter was to compare the predictive performance of the Cox neural network (Cox-nnet), the standard Cox PH model and the latent class regression Cox PH modelling. The results showed that the Cox-nnet does not offer any substantive improvement in predictive performance when compared to the standard Cox PH model. The Latent class regression approach offered a better predictive performance when compared to the Cox-nnet and the standard Cox PH model. Therefore, we concluded that ML Cox neural network does not offer improved prediction over standard statistical methods.

8.2 Limitations of current work and proposed further work

It has been demonstrated in Chapters 3, the benefits of simulating data that respects a causal process and how this aspect can help to facilitate prediction modelling. However, several caveats associated with this process must be noted. Due to several hypotheses being considered at the same time, more parameters are required when defining DGMs to simulate complex datasets. This in turn leads to problems such as model identification and/or model convergence.

Simulating data that follow a predefined causal structure is computationally intensive and, in some cases, algorithms may not converge. One possible way of addressing this issue is by reformulating the hypotheses and repeating the process until the model is identified and converges within acceptable timeframes.

The trade-off of computational investment and improved prediction is part of the challenges that warrants further exploration.

Due to the complexity in the DGM, prediction models generated for such data require more variables which might lead to over-parameterisation which may eventually lead to overfitting. On the other hand, as the number of parameters increase, bias decreases because the models are able to capture the underlying non-linear patterns more easily. However, complex models are likely to exhibit higher variance due to overfitting and therefore do not generalise to other datasets.

Model misspecification is another limitation associated with this approach. Most real-life studies contain complex relationships and patterns that are hard to hypothesise. Even though, using DGMs in such contexts can help reveal new causal mechanisms and lead to the development of new hypotheses, there is a chance that the simulated dataset may not fully capture whole ground truth, which may eventually introduce bias.

It has been demonstrated in Chapters 5 and 6 that LCR models can provide substantive improvements in predictive acuity and clinical utility over standard approaches using generalised linear models (GLM). Nonetheless, there are several potential limitations that warrant consideration and further investigation. Firstly, it would be insightful to compare these alternative approaches to prediction using larger datasets and larger numbers of covariates than those chosen for illustration in this thesis. This might involve comparing all models considered for Procedures $1-4$ in Chapters 5 using different numbers and sets of covariates from similar sized datasets as well as extending the application of LCR modelling to more complex scenarios and much larger datasets.

Our conclusions in Chapter 6 are somewhat preliminary since no subsequent iter-

ations of the models were attempted involving, for example, iterative selections of candidate predictors, alternative numbers of latent classes and so on and no subsequent calibration or evaluation of external validity was performed. While the approach adopted was intended to simplify the analyses summarised and facilitate comparisons between models generated using different modelling techniques in different scenarios (and with different numbers of continuous and dichotomised predictors). Further development of these models would be required to fully optimise (and evaluate) their performance and practical utility. Testing (and refining) such models is computationally resource-intensive and is beyond the scope of the present study, but much would be gained by further research on similar (simulated or real world) data, relevant to a range of scenarios in which the causal structure of the underlying data generated mechanism is either known (and fixed, through simulation) or well enough understood to facilitate interpretation (such as those where the mechanisms involved are subject to established physical laws, determined through design, or has been robustly interrogated through prior experimentation).

We explored the Machine learning's Cox-nnet model for survival prediction with different architectures and compared against the LCR modelling, from which we concluded that the LCR approach is better. It would also be worthy exploring whether novel architectures can be explored for easy comparison with the LCR modelling. Future work should explore a novel architecture with two nodes in the cox regression output layer of the Cox-nnet model to assess whether the apparent benefits of a 2 class LCR models might be easily replicated within the new Cox-nnet architecture.

Further analysis of the node structure of the Cox-nnet would also be insightful to

determine whether it can be compared with the latent classes obtained through a latent class regression model.

We also notice the absence of consistent improvements in performance between 1-class and 2-class models, and given the sizeable number of 2-class Cox PH models that failed to converge, it is not yet possible to recommend that 2-class Cox PH LCR models should always be used in preference to standard 1-class Cox PH models (or in which causal scenarios each is likely to perform best); and it is not yet possible to conclude whether differences in performance might offer reliable insights into underlying causal structures. However, our work suggests that such recommendations and insights may well emerge following more in-depth research and development which warrants further attention.

An important challenge with latent class modelling is its sensitivity to starting values, because these are used to maximise the likelihood function when estimating model parameters. Where the starting values are far from the optimum solution, the likelihood function takes longer to converge and may even fail to do so. Occasionally, up to 50% of the random starts chosen will generate meaningful solutions when the likelihood function is maximised. For a solution to be meaningful, the highest likelihood value is expected to be replicated many times. When this does not occur, it signifies that either no solution has been achieved and the number of random starts needs to be increased to converge on a global optimum solution or the specified model structure is unsuitable for the given dataset. While this can add to the time required to explore optimum solutions, once the target values are estimated they can be used as initial values for the final models derived, thereby reducing the duration of the final search process.

8.3 Conclusions and recommendations

Understanding and respecting the data generation process when conducting prediction modelling is important because it helps to avoid making incorrect inferences due to incorrect and inappropriate predictions. We therefore recommend that researchers should aim to use their theoretical understanding of the data generation process when making predictions for individuals or subgroups of patients, instead of merely applying 'big' data into the existing methods without paying attention to the underlying causal structure.

Appendix A

Supplementary details for Chapter 2

A.1 Partial loglikelihood for the Cox PH model

Let $\boldsymbol{X}^T = [X_1, X_2, \ldots, X_p]$ be a vector of p covariates or predictors for individual patients, $\boldsymbol{t} = [t_1, t_2, \ldots, t_n]$ be the survival times for the n patients. The Cox PH model can be defined as

$$h(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{\beta}) = h_0(\boldsymbol{t}) \exp(\boldsymbol{\beta}^T \boldsymbol{X}), \tag{A.1}$$

where $\boldsymbol{\beta}^T = [\beta_1, \beta_2, \ldots, \beta_p]$ is a vector of parameters and $h_0(\boldsymbol{t})$ is called the baseline hazard function, and reflects the underlying hazard when the effect of the covariates is equal to zero. In other words when $X_1, X_2, \ldots, X_p$ is equal to zero. Suppose that all the failure times are distinct (no ties), the probability that an event will happen at t_i given the number of individuals in a risk set, $\mathbb{R}(t_i)$ is given

by

$$P(t_i|\mathbb{R}(t_i)) = \frac{\exp(\boldsymbol{\beta}^T \boldsymbol{X_{(i)}})}{\sum_{j\in\mathbb{R}(t_i)} \exp(\boldsymbol{\beta}^T \boldsymbol{X_j})} \tag{A.2}$$

where $\mathbb{R}(t_i)$ is the set of all subjects that are at risk of experiencing the event at that point $(t = t_i)$.

The partial likelihood function is given as a product over the observed failure times of conditional probabilities, of observing an event of interest, given the risk set at that time.

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} \left[\frac{\exp(\boldsymbol{\beta}^T \boldsymbol{X}_{(i)})}{\sum_{j\in\mathbb{R}_i} \exp(\boldsymbol{\beta}^T \boldsymbol{X}_j)} \right]^{\sigma_i} \tag{A.3}$$

where

$$\sigma_i = \begin{cases} 1, & \text{if an event occurred} \\ 0, & \text{if an event did not occur} \end{cases}$$

Taking logs of both sides of equation A.3 yields

$$\log L(\boldsymbol{\beta}) = \log \left\{ \prod_{i=1}^{n} \left[\frac{\exp(\boldsymbol{\beta}^T \boldsymbol{X}_{(i)})}{\sum_{j \in \mathbb{R}_i} \exp(\boldsymbol{\beta}^T \boldsymbol{X}_j)} \right]^{\sigma_i} \right\}$$

$$= \log \left\{ \left[\frac{\exp(\boldsymbol{\beta}^T \boldsymbol{X}_{(1)})}{\sum_{j \in \mathbb{R}_i} \exp(\boldsymbol{\beta}^T \boldsymbol{X}_j)} \right]^{\sigma_1} \times \ldots \times \left[\frac{\exp(\boldsymbol{\beta}^T \boldsymbol{X}_{(n)})}{\sum_{j \in \mathbb{R}_i} \exp(\boldsymbol{\beta}^T \boldsymbol{X}_j)} \right]^{\sigma_n} \right\}$$
$$= \log \left[\frac{\exp(\boldsymbol{\beta}^T \boldsymbol{X}_{(1)})}{\sum_{j \in \mathbb{R}_i} \exp(\boldsymbol{\beta}^T \boldsymbol{X}_j)} \right]^{\sigma_1} + \ldots + \log \left[\frac{\exp(\boldsymbol{\beta}^T \boldsymbol{X}_{(n)})}{\sum_{j \in \mathbb{R}_i} \exp(\boldsymbol{\beta}^T \boldsymbol{X}_j)} \right]^{\sigma_n}$$

$$= \sigma_1 \log \left[\frac{\exp(\boldsymbol{\beta}^T \boldsymbol{X}_{(1)})}{\sum_{j \in \mathbb{R}_i} \exp(\boldsymbol{\beta}^T \boldsymbol{X}_j)} \right] + \ldots + \sigma_n \log \left[\frac{\exp(\boldsymbol{\beta}^T \boldsymbol{X}_{(n)})}{\sum_{j \in \mathbb{R}_i} \exp(\boldsymbol{\beta}^T \boldsymbol{X}_j)} \right]$$
$$= \sigma_1 \left[\boldsymbol{\beta}^T \boldsymbol{X}_{(1)} - \log \sum_{j \in \mathbb{R}_i} \exp(\boldsymbol{\beta}^T \boldsymbol{X}_j) \right] + \ldots + \sigma_n \left[\boldsymbol{\beta}^T \boldsymbol{X}_{(n)} - \log \sum_{j \in \mathbb{R}_i} \exp(\boldsymbol{\beta}^T \boldsymbol{X}_j) \right]$$
$$= \sum_{i=1}^{n} \sigma_i \boldsymbol{\beta}^T \boldsymbol{X}_{(i)} - \sum_{i=1}^{n} \sigma_i \log \sum_{j \in \mathbb{R}_i} \exp(\boldsymbol{\beta}^T \boldsymbol{X}_j)$$
$$= \sum_{i=1}^{n} \sigma_i \left[\boldsymbol{\beta}^T \boldsymbol{X}_{(i)} - \log \sum_{j \in \mathbb{R}_i} \exp(\boldsymbol{\beta}^T \boldsymbol{X}_j) \right] \tag{A.4}$$

The partial-log likelihood function is given by

$$PLL(\boldsymbol{\beta}) = \sum_{i=1}^{n} \sigma_i \left[\boldsymbol{\beta}^T \boldsymbol{X}_{(i)} - \log \sum_{j \in \mathbb{R}_i} \exp(\boldsymbol{\beta}^T \boldsymbol{X}_j) \right] \tag{A.5}$$

A.2 Partial loglikelihood for the Cox-nnet model

In a Cox-nnet model, the covariate vector is replaced by the output of the hidden layer yielding a modified PLL defined as

$$PLL(\boldsymbol{\beta}, W) = \sum_{i=1}^{n} \sigma_i \left[\boldsymbol{\beta}^T \phi(\boldsymbol{W^T X_{(i)}} + \boldsymbol{b}) - \log \sum_{j \in \mathbb{R}_i} \exp(\boldsymbol{\beta}^T \phi(\boldsymbol{W^T X_j} + \boldsymbol{b})) \right] \tag{A.6}$$

Subsequently, the partial-log likelihood in A.6 is extended by adding a ridge regularisation term yielding a cost function $C(\boldsymbol{\beta}, W)$ which is minimised through back propagation.

$$C(\boldsymbol{\beta}, W) = PLL(\boldsymbol{\beta}, W) + \lambda(\|\boldsymbol{\beta}\|_2 + \|\boldsymbol{W}\|_2) \tag{A.7}$$

where

- σ_i is the censoring indicator for patient i.
- $\boldsymbol{\beta}^T$ is a vector for the regression coefficients.
- $\boldsymbol{W}$ is the coefficient weight matrix between the input and the hidden layer.
- $\boldsymbol{b}$ is the bias term for each hidden node.
- $\boldsymbol{X}_{(i)}$ is the covariate vector for patient i.
- $\phi(.)$ is the tanh activation function as shown in equation 2.15 and it is applied element-wise on a vector.
- λ is the regularisation parameter.

- $\|.\|$ is the L^2 norm. The L^2 norm is calculated as the square root of the sum of the squared vector values, e.g. Let $\boldsymbol{X}$=(x_1, x_2, x_3), $\|\boldsymbol{X}\| = \sqrt{{x_1}^2 + {x_2}^2 + {x_3}^2}$
- $\mathbb{R}_i$ is the risk set. A risk set is defined as the set of individuals at risk of experiencing an event of interest at a particular timepoint.
- $\boldsymbol{X}_j$ is the covariate vector for the patients in the risk set.
- $\phi(\boldsymbol{W^T X_{(i)} + b})$ is the output for patient i
- $\phi(\boldsymbol{W^T X_j + b})$ is the output for patient j where $j \in \mathbb{R}_i$

A.3 An illustration of the risk set

Consider a sample dataset with 4 variables namely: Patient Id, survival time, censoring indicator and Age of the patients. We first of all rank the patients according to their survival time.

Table A.1: A sample dataset

ID	Time	Indicator	Age
1	5.37	1	75
2	7.78	1	72
3	2.05	1	70
4	9.25	0	55
5	8.85	1	68
6	2.25	1	76

In Table A.2 below we have created a column called risk set which has the set of individuals at risk of experiencing an event of interest at each timepoint. The

patient with ID number of 3 was the first to experience the event. The number of individuals in the risk set at the time the patient with ID = 3 experienced the event is 6 (i.e. the risk set is $1, 2, 3, 4, 5, 6$). The next patient to experience the event is patient with ID = 6. The risk set at this point is $1, 2, 4, 5, 6$. The risk set does not include the patient with ID = 3 because this patient is no longer in the study. The number of patients keeps on reducing until there is only one patient remaining in the risk set.

The last column shows the contribution of each patient to the partial likelihood

Table A.2: Risk set and likelihood contribution

Time	Indicator	Age	ID	Risk set	Likelihood
2.05	1	70	3	{1,2,3,4,5,6}	$\frac{e^{(70\beta)}}{e^{(75\beta)}+e^{(72\beta)}+e^{(70\beta)}+e^{(55\beta)}+e^{(68\beta)}+e^{(76\beta)}}$
2.25	1	76	6	{1,2,4,5,6}	$\frac{e^{(76\beta)}}{e^{(75\beta)}+e^{(72\beta)}+e^{(55\beta)}+e^{(68\beta)}+e^{(76\beta)}}$
5.37	1	75	1	{1,2,4,5}	$\frac{e^{(75\beta)}}{e^{(75\beta)}+e^{(72\beta)}+e^{(55\beta)}+e^{(68\beta)}}$
7.78	1	72	2	{2,4,5}	$\frac{e^{(72\beta)}}{e^{(72\beta)}+e^{(55\beta)}+e^{(68\beta)}}$
8.85	1	68	5	{4,5}	$\frac{e^{(68\beta)}}{e^{(55\beta)}+e^{(68\beta)}}$
9.25	0	55	4	{4}	1

The partial likelihood function is given as a product over the observed failure times of conditional probabilities, of observing an event of interest, given the risk set at that time. The product of the terms in the last column in Table A.2 is what gives us the partial likelihood function. The column for the likelihood shows each individual's contribution to the likelihood. The risk set introduces dependency between the terms. As such, it is meaningless to evaluate partial sums, hence, it is impossible to have mini batches or to do stochastic gradient descent.

Appendix B

Supplementary details for Chapter 4

B.1 Change-score simulation R-code

```
rm(list=ls())
if( packageVersion('dagitty')<"0.2.3" ){
                warning("Please install at least version
                   0.2.3 of the dagitty package!")

        stop("Use this command:
             devtools::install_github('jtextor/dagitty/r')") }

require(dagitty); require(MASS); require(rpsychi);
 library(matrixcalc); library(tidyverse); library(dplyr)

##############################
## Define various functions ##
##############################

# function to summarise best fitting model
Val <- function(txt) {
  Code <- sum(9000,grep("X0",txt)*100,
  grep("U0",txt)*10, grep("Y0",txt),na.rm=TRUE)
```

```
  return(Code) }

getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))] }

# function to execute multiple simulations and summarise findings
runSim <- function(Mu,Sigma,Nobs,Nreps) {
  if (!is.positive.definite(round(Sigma,8))) {
    Sum <- matrix(rep(0,3*24),nrow=3,ncol=24)
    colnames(Sum) <- c("Y1","DY","Y0","X0","U0","Y1"
       ,"DY","Y0","X0" ,"U0","Y0_Y1","Y0_U0"
        ,"Y0_X0","Y0_DY" ,"Y1_U0","Y1_X0","Y1_DY","U0_X0",
                       "U0_DY","X0_DY","Y1_RMSE","DY_RMSE",
                              "Y1_XUY","DY_XUY")
  } else {
    Sum   <- NULL
    for (itn in 1:Nreps) {
      # simulate a single dataset & calculate change score
         & summary information
      dat        <- data.frame(mvrnorm(Nobs,Mu,Sigma,empirical=FALSE));
      names(dat) <-c("Y0","Y1","U0","X0")
      dat$DY     <- dat$Y1 - dat$Y0
      MyData     <- select(tibble(data.frame(dat)),Y1,DY,Y0,X0,U0)
      Means      <- apply(MyData,2,mean)
      SDs        <- apply(MyData,2,sd)
      CorMat     <- cor(MyData)
      Corrs      <- CorMat[lower.tri(CorMat)]
      names(Corrs) <- c("Y0_Y1","Y0_U0","Y0_X0","Y0_DY","Y1_U0","Y1_X0"
      ,"Y1_DY","U0_X0","U0_DY","X0_DY")

      # create Train & Test datasets
      Select     <- sample(c(1:Nobs),Nobs*0.7,replace=FALSE)
      Train      <- MyData[Select,]
      Test       <- MyData[-Select,]

      # generate list of prediction covariates & lm formulae ignoring
Y0
      Vars       <- as.character(names(MyData)[-c(1:3)])
      input      <- expand.grid(data.frame(matrix(rep(c(TRUE,FALSE)
          ,length(Vars)),nrow=2)))
      Y1.Form    <- apply(input,1,function(x)
      {as.formula(paste(c("Y1 ~ 1",Vars[x]),collapse = "+"))} )
```

```
      DY.Form    <- apply(input,1,function(x)
      {as.formula(paste(c("DY ~ 1",Vars[x]),collapse = "+"))} )

      # run lm on Train data & calculate RMSEs on Test data
      Y1.BIC     <- sapply(Y1.Form,function(x)
      {BIC(lm(x,data=Train))} )
      DY.BIC     <- sapply(DY.Form,function(x)
      {BIC(lm(x,data=Train))} )
      Y1.Best    <- Reduce(paste,deparse(Y1.Form[[which(Y1.BIC
                                ==min(Y1.BIC))]]))
      DY.Best    <-Reduce(paste,deparse(DY.Form[[which(DY.BIC
                                 ==min(DY.BIC))]]))
      Y1.lm      <- lm(Y1.Best,data=Train)
      DY.lm      <- lm(DY.Best,data=Train)
      Y1.pred    <- predict(Y1.lm,Test)
      DY.pred    <- predict(DY.lm,Test)
      Y1.RMSE    <- sqrt(mean((Test$Y1-Y1.pred)^2))
      DY.RMSE    <- sqrt(mean((Test$DY-DY.pred)^2))
      Best       <- c(Val(Y1.Best),Val(DY.Best));
       names(Best) <- c("Y1_XUY","DY_XUY")
      RMSE       <- c(Y1.RMSE,DY.RMSE);
       names(RMSE) <- c("Y1_RMSE","DY_RMSE")
      Sum        <- rbind(Sum,c(Means,SDs,Corrs,RMSE,Best))
    }
  }
  return(Sum) }

DAG_con_con_con <- function(pU0_X0,pU0_Y0,pU0_Y1,pX0_Y0,pX0_Y1,pY0_Y1) {
# X0 -> Y0 / U0 -> X0 / U0 -> Y0 #
  dag <- dagitty(paste0("dag{ X0->Y0 [beta=",pX0_Y0,"]
        U0->X0 [beta=",pU0_X0,"]  U0->Y0 [beta=",pU0_Y0,"]
                              U0->Y1 [beta=",pU0_Y1,"]  Y0->Y1
                              [beta=",pY0_Y1,"]  X0->Y1
                              [beta=",pX0_Y1,"] }"))
  return(dag) }

DAG_con_con_med <- function(pU0_X0,pU0_Y0,pU0_Y1,pX0_Y0,pX0_Y1,pY0_Y1) {
 # X0 -> Y0 / U0 -> X0 / Y0 -> U0 #
  dag <- dagitty(paste0("dag{ X0->Y0 [beta=",pX0_Y0,"]
       U0->X0 [beta=",pU0_X0,"]  Y0->U0 [beta=",pU0_Y0,"]
                              U0->Y1 [beta=",pU0_Y1,"]  Y0->Y1
                              [beta=",pY0_Y1,"]  X0->Y1
                              [beta=",pX0_Y1,"] }"))
```

```
  return(dag) }

DAG_med_con_con <- function(pU0_X0,pU0_Y0,pU0_Y1,pX0_Y0,pX0_Y1,pY0_Y1){
 # Y0 -> X0 / U0 -> X0 / U0 -> Y0 #
  dag <- dagitty(paste0("dag{ Y0->X0 [beta=",pX0_Y0,"]
  U0->X0 [beta=",pU0_X0,"]  U0->Y0 [beta=",pU0_Y0,"]
                              U0->Y1 [beta=",pU0_Y1,"]  Y0->Y1
                              [beta=",pY0_Y1,"]  X0->Y1
                              [beta=",pX0_Y1,"] }"))
  return(dag) }

DAG_med_con_med <- function(pU0_X0,pU0_Y0,pU0_Y1,pX0_Y0,pX0_Y1,pY0_Y1){
# Y0 -> X0 / U0 -> X0 / Y0 -> U0 #
  dag <- dagitty(paste0("dag{ Y0->X0 [beta=",pX0_Y0,"]
  U0->X0 [beta=",pU0_X0,"]  Y0->U0 [beta=",pU0_Y0,"]
                              U0->Y1 [beta=",pU0_Y1,"]  Y0->Y1
                              [beta=",pY0_Y1,"]  X0->Y1
                              [beta=",pX0_Y1,"] }"))
  return(dag)}

DAGsim <- function(dag,Nmu,Sigma,Nobs,Nreps) {
  Cor     <- impliedCovarianceMatrix(dag)
  Sigma   <- r2cov(sqrt(Nvar),Cor)
  Sim     <- runSim(Nmu,Sigma,Nobs,Nreps)
  return(Sim) }

##############################################
## Simulates scenarios for orthogonal X0-Y0 ##
##############################################

RCTSim <- function(Nmu,Sigma,Nobs,Nreps,Y0Y1Seq,X0Y1Seq,U0Y1Seq) {

  start   <- Sys.time()
  # set consistent path coefficient
  pX0_Y0  <- 0.0
  Summ    <- Mode <- NULL
  Nruns   <- length(Y0Y1Seq)*length(X0Y1Seq)*length(U0Y1Seq)*7
  Step    <- 1; PctProg <- round(100*(Step/Nruns),2)
  for (pY0_Y1 in Y0Y1Seq) {
    for (pX0_Y1 in X0Y1Seq) {
      for (pU0_Y1 in U0Y1Seq) {
```

```
## X0-Y0 orthogonal & no U0 confounding
pX0_Y0  <- pU0_X0  <- pU0_Y0 <- 0
dag     <- DAG_con_con_con(pU0_X0,pU0_Y0,pU0_Y1,pX0_Y0,
                                        pX0_Y1,pY0_Y1)

Config  <- data.frame(pY0_Y1=pY0_Y1, pX0_Y1=pX0_Y1,
              pU0_Y1=pU0_Y1, pU0_Y0=pU0_Y0, pX0_Y0=pX0_Y0,
               pU0_X0=pU0_X0, U0_Y0=1, X0_Y0=1)
Sim     <- DAGsim(dag,Nmu,Sigma,Nobs,Nreps)
Mode    <- rbind(Mode,cbind(t(apply(Sim[,21:22],2,getmode)),
                    Config))
Summ    <- rbind(Summ,cbind(apply(Sim[,1:20],2,function(x)
     {quantile(x,c(0.025,0.5,0.975))}),Config))
print(c(as.numeric(as.character(Config)),PctProg))
Step    <- Step + 1; PctProg <- round(100*(Step/Nruns),2)

## X0-Y0 orthogonal & U0 confounds X0

pX0_Y0  <- pU0_Y0 <- 0
for (pU0_X0 in c(-0.5,0.5)) {
  dag     <- DAG_con_con_con(pU0_X0,pU0_Y0,pU0_Y1,pX0_Y0
                      ,pX0_Y1,pY0_Y1)
  Config  <- data.frame(pY0_Y1=pY0_Y1, pX0_Y1=pX0_Y1
                            , pU0_Y1=pU0_Y1, pU0_Y0=pU0_Y0,
                               pX0_Y0=pX0_Y0, pU0_X0=pU0_X0,
                               U0_Y0=1, X0_Y0=1)
  Sim     <- DAGsim(dag,Nmu,Sigma,Nobs,Nreps)
  Mode    <- rbind(Mode,cbind(t(apply(Sim[,21:22],
                                2,getmode)),Config))
  Summ    <- rbind(Summ,cbind(apply(Sim[,1:20],
               2,function(x) {quantile(x,c(0.025,0.5,0.975))}),
                             Config))
  print(c(as.numeric(as.character(Config)),PctProg))
  Step    <- Step + 1; PctProg <- round(100*(Step/Nruns),2) }

## X0-Y0 orthogonal & U0 confounds Y0
pX0_Y0 <- pU0_X0 <- 0
for (pU0_Y0 in c(-0.5,0.5)) {
  dag     <- DAG_con_con_con(pU0_X0,pU0_Y0,pU0_Y1,pX0_Y0
  ,pX0_Y1,pY0_Y1)
  Config  <- data.frame(pY0_Y1=pY0_Y1, pX0_Y1=pX0_Y1
       , pU0_Y1=pU0_Y1, pU0_Y0=pU0_Y0,
          pX0_Y0=pX0_Y0, pU0_X0=pU0_X0, U0_Y0=1,
```

```
                X0_Y0=1)
          Sim      <- DAGsim(dag,Nmu,Sigma,Nobs,Nreps)
          Mode     <- rbind(Mode,cbind(t(apply(Sim[,21:22],
                                2,getmode)),Config))
          Summ     <- rbind(Summ,cbind(apply(Sim[,1:20],
                             2,function(x) {quantile(x,c(0.025,0.5,0.975))}),
                                Config))
          print(c(as.numeric(as.character(Config)),PctProg))
          Step     <- Step + 1; PctProg <- round(100*(Step/Nruns),2) }

        ## X0-Y0 orthogonal & U0 mediates Y0
        pX0_Y0 <- pU0_X0 <- 0
        for (pU0_Y0 in c(-0.5,0.5)) {
          dag      <- DAG_con_con_med(pU0_X0,pU0_Y0,pU0_Y1,pX0_Y0,
                pX0_Y1,pY0_Y1)
          Config   <- data.frame(pY0_Y1=pY0_Y1,
                    pX0_Y1=pX0_Y1, pU0_Y1=pU0_Y1,
                     pU0_Y0=pU0_Y0, pX0_Y0=pX0_Y0,
                     pU0_X0=pU0_X0, U0_Y0=0, X0_Y0=1)
          Sim      <- DAGsim(dag,Nmu,Sigma,Nobs,Nreps)
          Mode     <- rbind(Mode,cbind(t(apply(Sim[,21:22],
                              2,getmode)),Config))
          Summ     <- rbind(Summ,cbind(apply(Sim[,1:20],
                               2,function(x) {quantile(x,c(0.025,0.5,0.975))})
                              ,Config))
          print(c(as.numeric(as.character(Config)),PctProg))
          Step     <- Step + 1; PctProg <- round(100*(Step/Nruns),2) }
      }
    }
  }
  end        <- Sys.time();  print(end-start)
  return(list(Mode,Summ))
}

##############################################
## Simulates scenarios for X0 confounds Y0 ##
##############################################

ConSim <- function(Nmu,Sigma,Nobs,Nreps,Y0Y1Seq,X0Y1Seq,U0Y1Seq) {

  start    <- Sys.time()
  Summ     <- Mode <- NULL
  Nruns    <- length(Y0Y1Seq)*length(X0Y1Seq)*length(U0Y1Seq)*2*9
```

```
Step    <- 1; PctProg <- round(100*(Step/Nruns),2)
for (pY0_Y1 in Y0Y1Seq) {
  for (pX0_Y1 in X0Y1Seq) {
    for (pU0_Y1 in U0Y1Seq) {
      for (pX0_Y0 in c(-0.5,0.5)) {

        ## X0 confounds Y0 & no U0 confounding
        pU0_X0  <- pU0_Y0 <- 0
        dag     <- DAG_con_con_con(pU0_X0,pU0_Y0,pU0_Y1,pX0_Y0
              ,pX0_Y1,pY0_Y1)
        Config  <- data.frame(pY0_Y1=pY0_Y1, pX0_Y1=pX0_Y1,
                    pU0_Y1=pU0_Y1, pU0_Y0=pU0_Y0,
                     pX0_Y0=pX0_Y0, pU0_X0=pU0_X0,
                       U0_Y0=1, X0_Y0=1)
        Sim     <- DAGsim(dag,Nmu,Sigma,Nobs,Nreps)
        Mode    <- rbind(Mode,cbind(t(apply(Sim[,21:22],
                          2,getmode)),Config))
        Summ    <- rbind(Summ,cbind(apply(Sim[,1:20],
                       2,function(x) {quantile(x,c(0.025,0.5,0.975))}),
                             Config))
        print(c(as.numeric(as.character(Config)),PctProg))
        Step    <- Step + 1; PctProg <- round(100*(Step/Nruns),2)

        ## X0 confounds Y0 & U0 confounds X0

        pU0_Y0 <- 0
        for (pU0_X0 in c(-0.5,0.5)) {

          dag     <- DAG_con_con_con(pU0_X0,pU0_Y0,pU0_Y1,
                           pX0_Y0,pX0_Y1,pY0_Y1)
          Config  <- data.frame(pY0_Y1=pY0_Y1,pX0_Y1=pX0_Y1,
                       pU0_Y1=pU0_Y1, pU0_Y0=pU0_Y0,
                         pX0_Y0=pX0_Y0, pU0_X0=pU0_X0,
                           U0_Y0=1, X0_Y0=1)
          Sim     <- DAGsim(dag,Nmu,Sigma,Nobs,Nreps)
          Mode    <- rbind(Mode,cbind(t(apply(Sim[,21:22],
                                2,getmode)),Config))
          Summ    <- rbind(Summ,cbind(apply(Sim[,1:20],
          2,function(x) {quantile(x,c(0.025,0.5,0.975))}),Config))
          print(c(as.numeric(as.character(Config)),PctProg))
          Step    <- Step + 1;
          PctProg <- round(100*(Step/Nruns),2)}
```

```
            ## X0 confounds Y0 & U0 confounds Y0
            pU0_X0 <- 0
            for (pU0_Y0 in c(-0.5,0.5)) {
              dag      <- DAG_con_con_con(pU0_X0,pU0_Y0,pU0_Y1,pX0_Y0,
                   pX0_Y1,pY0_Y1)
              Config   <- data.frame(pY0_Y1=pY0_Y1,
                    pX0_Y1=pX0_Y1, pU0_Y1=pU0_Y1
                        , pU0_Y0=pU0_Y0, pX0_Y0=pX0_Y0,
                           pU0_X0=pU0_X0, U0_Y0=1, X0_Y0=1)
              Sim      <- DAGsim(dag,Nmu,Sigma,Nobs,Nreps)
              Mode     <- rbind(Mode,cbind(t(apply(Sim[,21:22],
                                2,getmode)),Config))
              Summ     <- rbind(Summ,cbind(apply(Sim[,1:20],
                   2,function(x) {quantile(x,c(0.025,0.5,0.975))})
                                    ,Config))
              print(c(as.numeric(as.character(Config)),PctProg))
              Step     <- Step + 1; PctProg <- round(100*(Step/Nruns),2)}

            ## X0 confounds Y0 & U0 confounds X0 & U0 confounds Y0
            for (pU0_X0 in c(-0.5,0.5)) {
              for (pU0_Y0 in c(-0.5,0.5)) {
                dag      <- DAG_con_con_con(pU0_X0,pU0_Y0,pU0_Y1,pX0_Y0,
                    pX0_Y1,pY0_Y1)
                Config   <- data.frame(pY0_Y1=pY0_Y1,
                 pX0_Y1=pX0_Y1, pU0_Y1=pU0_Y1,
                  pU0_Y0=pU0_Y0, pX0_Y0=pX0_Y0,
                   pU0_X0=pU0_X0, U0_Y0=1, X0_Y0=1)
                Sim      <- DAGsim(dag,Nmu,Sigma,Nobs,Nreps)
                Mode     <- rbind(Mode,cbind(t(apply(Sim[,21:22],
                                  2,getmode)),Config))
                Summ     <- rbind(Summ,cbind(apply(Sim[,1:20],
                            2,function(x) {quantile(x,c(0.025,0.5,0.975))})
                                          ,Config))
                print(c(as.numeric(as.character(Config)),PctProg))
                Step     <- Step + 1;
                PctProg <- round(100*(Step/Nruns),2)}}
          }
        }
      }
    }
    end        <- Sys.time();  print(end-start)
    return(list(Mode,Summ))
}
```

```
##############################################
## Simulates scenarios for X0 mediates Y0 ##
##############################################

MedSim <- function(Nmu,Sigma,Nobs,Nreps,Y0Y1Seq,X0Y1Seq,U0Y1Seq)
{

  start    <- Sys.time()
  Summ     <- Mode <- NULL
  Nruns    <- length(Y0Y1Seq)*length(X0Y1Seq)*length(U0Y1Seq)*2*9
  Step     <- 1; PctProg <- round(100*(Step/Nruns),2)
  for (pY0_Y1 in Y0Y1Seq) {
    for (pX0_Y1 in X0Y1Seq) {
      for (pU0_Y1 in U0Y1Seq) {
        for (pX0_Y0 in c(-0.5,0.5)) {

          ## X0 mediates Y0 & no U0 confounding
          pU0_X0  <- pU0_Y0 <- 0
          dag     <- DAG_med_con_con(pU0_X0,pU0_Y0,pU0_Y1,
                          pX0_Y0,pX0_Y1,pY0_Y1)
          Config  <- data.frame(pY0_Y1=pY0_Y1, pX0_Y1=pX0_Y1,
                            pU0_Y1=pU0_Y1, pU0_Y0=pU0_Y0,
                              pX0_Y0=pX0_Y0, pU0_X0=pU0_X0,
                                  U0_Y0=1, X0_Y0=0)
          Sim     <- DAGsim(dag,Nmu,Sigma,Nobs,Nreps)
          Mode    <- rbind(Mode,cbind(t(apply(Sim[,21:22],
                                  2,getmode)),Config))
          Summ    <- rbind(Summ,cbind(apply(Sim[,1:20],
                        2,function(x) {quantile(x,c(0.025,0.5,0.975))})
                                  ,Config))
          print(c(as.numeric(as.character(Config)),PctProg))
          Step    <- Step + 1; PctProg <- round(100*(Step/Nruns),2)

          ## X0 mediates Y0 & U0 confounds X0
          pU0_Y0 <- 0
          for (pU0_X0 in c(-0.5,0.5)) {
            dag     <- DAG_med_con_con(pU0_X0,pU0_Y0,pU0_Y1,pX0_Y0,
                          pX0_Y1,pY0_Y1)
            Config  <- data.frame(pY0_Y1=pY0_Y1,
                pX0_Y1=pX0_Y1, pU0_Y1=pU0_Y1,
                pU0_Y0=pU0_Y0, pX0_Y0=pX0_Y0,
                   pU0_X0=pU0_X0, U0_Y0=1, X0_Y0=0)
```

```
  Sim      <- DAGsim(dag,Nmu,Sigma,Nobs,Nreps)
  Mode     <- rbind(Mode,cbind(t(apply(Sim[,21:22],
              2,getmode)),Config))
  Summ     <- rbind(Summ,cbind(apply(Sim[,1:20],
             2,function(x) {quantile(x,c(0.025,0.5,0.975))})
                 ,Config))
  print(c(as.numeric(as.character(Config)),PctProg))
  Step     <- Step + 1;
    PctProg <- round(100*(Step/Nruns),2) }

## X0 mediates Y0 & U0 confounds Y0
pU0_X0 <- 0
for (pU0_Y0 in c(-0.5,0.5)) {
  dag      <- DAG_med_con_con(pU0_X0,pU0_Y0,pU0_Y1,
                              pX0_Y0,pX0_Y1,pY0_Y1)
  Config   <- data.frame(pY0_Y1=pY0_Y1,
                pX0_Y1=pX0_Y1,pU0_Y1=pU0_Y1,
                    pU0_Y0=pU0_Y0, pX0_Y0=pX0_Y0,
                    pU0_X0=pU0_X0, U0_Y0=1, X0_Y0=0)
  Sim      <- DAGsim(dag,Nmu,Sigma,Nobs,Nreps)
  Mode     <- rbind(Mode,cbind(t(apply(Sim[,21:22],
                2,getmode)),Config))
  Summ     <- rbind(Summ,cbind(apply(Sim[,1:20],
             2,function(x) {quantile(x,c(0.025,0.5,0.975))})
                             ,Config))
  print(c(as.numeric(as.character(Config)),PctProg))
  Step     <- Step + 1;
  PctProg <- round(100*(Step/Nruns),2) }

## X0 mediates Y0 & U0 confounds X0 & U0 confounds Y0
for (pU0_X0 in c(-0.5,0.5)) {
  for (pU0_Y0 in c(-0.5,0.5)) {
    dag      <- DAG_med_con_med(pU0_X0,pU0_Y0,pU0_Y1,pX0_Y0,
                    pX0_Y1,pY0_Y1)
    Config   <- data.frame(pY0_Y1=pY0_Y1,
           pX0_Y1=pX0_Y1, pU0_Y1=pU0_Y1,
               pU0_Y0=pU0_Y0, pX0_Y0=pX0_Y0,
                     pU0_X0=pU0_X0, U0_Y0=1, X0_Y0=0)
    Sim      <- DAGsim(dag,Nmu,Sigma,Nobs,Nreps)
    Mode     <- rbind(Mode,cbind(t(apply(Sim[,21:22],
                     2,getmode)),Config))
    Summ     <- rbind(Summ,cbind(apply(Sim[,1:20],
           2,function(x) {quantile(x,c(0.025,0.5,0.975))})
```

```
                                        ,Config))
                print(c(as.numeric(as.character(Config)),PctProg))
                Step    <- Step + 1;
                          PctProg <- round(100*(Step/Nruns),2)} }
          }
        }
      }
    }
    end        <- Sys.time();  print(end-start)
    return(list(Mode,Summ))
}

#####################################################
## Set the simulation configuration parameters ##
#####################################################

# scale of simulations
Nobs    <- 1000                 # vary study sizes
Nreps   <- 100                  # choose appropriate number
                                        #  of repeated simulations for each
                                        #  scenario

# means and variances
U0mu    <- 10
U0var   <- 1.5^2
X0mu    <- 10
X0var   <- 1.5^2
Y0mu    <- 10
Y0var   <- 1.5^2
Y1mu    <- 10
Y1var   <- 1.5^2

# vectors for simulations
Nmu     <- c(Y0mu, Y1mu, U0mu, X0mu)
Nvar    <- c(Y0var,Y1var,U0var,X0var)
Mode    <- Summ <- NULL

# baseline outcome serial correlation path coefficient
Y0Y1Seq <- seq(0.05,0.95,0.05) #; Y0Y1Seq <-
                      seq(0.05,0.95,0.15)

# main covariate effect size path coefficient
X0Y1Seq <- c(seq(-0.95,-0.05,0.05),Y0Y1Seq) #; X0Y1Seq <-
```

```
                    c(seq(-0.95,-0.05,0.15),Y0Y1Seq)

# confounding options
U0Y1Seq <- c(-0.5,0.5)

#####################################################################
## loop key path coefficients - in 3 part for parallel execution ##
#####################################################################

# orthogonal X0-Y0
set.seed(13)
ResRCT     <- RCTSim(Nmu,Sigma,Nobs,Nreps,Y0Y1Seq,
                              X0Y1Seq,U0Y1Seq)
Mode       <- ResRCT[[1]]
Summ       <- ResRCT[[2]]
NameRCT1   <- paste0("Mode_RCT_Ignore_",Nreps,"reps_",
                 (Nobs/1000),"kobs_",substr(Sys.time(),1,10),".csv")
NameRCT2   <- paste0("Summ_RCT_Ignore_",Nreps,"reps_",
                (Nobs/1000),"kobs_",substr(Sys.time(),1,10),".csv")
write.table(t(colnames(Mode)),NameRCT1,sep=",",row.names=
                            TRUE,col.names=FALSE)
write.table(signif(Mode),NameRCT1,sep=",",row.names=
                          TRUE,col.names=FALSE,append=TRUE)
write.table(t(colnames(Summ)),NameRCT2,sep=",",row.names=
                        TRUE,col.names=FALSE)
write.table(signif(Summ),NameRCT2,sep=",",row.names=
                TRUE,col.names=FALSE,append=TRUE)

# X0 confounds Y0
set.seed(13+length(Y0Y1Seq)*length(X0Y1Seq)*length(U0Y1Seq)*7)
ResCon     <- ConSim(Nmu,Sigma,Nobs,Nreps,Y0Y1Seq,X0Y1Seq,U0Y1Seq)
Mode       <- ResCon[[1]]
Summ       <- ResCon[[2]]
NameCon1   <- paste0("Mode_Con_Ignore_",Nreps,"reps_",
              (Nobs/1000),"kobs_",substr(Sys.time(),1,10),".csv")
NameCon2   <- paste0("Summ_Con_Ignore_",Nreps,"reps_",
      (Nobs/1000),"kobs_",substr(Sys.time(),1,10),".csv")
write.table(t(colnames(Mode)),NameCon1,sep=",",row.names=
                TRUE,col.names=FALSE)
write.table(signif(Mode),NameCon1,sep=",",row.names=
             TRUE,col.names=FALSE,append=TRUE)
write.table(t(colnames(Summ)),NameCon2,sep=",",row.names=
                    TRUE,col.names=FALSE)
```

```
write.table(signif(Summ),NameCon2,sep=",",row.names=
                    TRUE,col.names=FALSE,append=TRUE)

# X0 mediates Y0
set.seed(13+length(Y0Y1Seq)*length(X0Y1Seq)*
length(U0Y1Seq)*7+length(Y0Y1Seq)*length(X0Y1Seq)*length(U0Y1Seq)*2*9)
ResMed     <- MedSim(Nmu,Sigma,Nobs,Nreps,Y0Y1Seq,X0Y1Seq,U0Y1Seq)
Mode       <- ResMed[[1]]
Summ       <- ResMed[[2]]
NameMed1   <- paste0("Mode_Med_Ignore_",Nreps,"reps_",
              (Nobs/1000),"kobs_",substr(Sys.time(),1,10),".csv")
NameMed2   <- paste0("Summ_Med_Ignore_",Nreps,"reps_",
                (Nobs/1000),"kobs_",substr(Sys.time(),1,10),".csv")
write.table(t(colnames(Mode)),NameMed1,sep=",",row.names=
                    TRUE,col.names=FALSE)
write.table(signif(Mode),NameMed1,sep=",",row.names=
                         TRUE,col.names=FALSE,append=TRUE)
write.table(t(colnames(Summ)),NameMed2,sep=",",row.names=
                   TRUE,col.names=FALSE)
write.table(signif(Summ),NameMed2,sep=",",row.names=
                     TRUE,col.names=FALSE,append=TRUE)

######################
## Combine results ##
######################

if (FALSE) {
  # final output file identifier
  NameRCT1   <- paste0("Mode_RCT_Ignore_",Nreps,"reps_",
                (Nobs/1000),"kobs_",substr(Sys.time(),1,10),".csv")
  NameRCT2   <- paste0("Summ_RCT_Ignore_",Nreps,"reps_",
                  (Nobs/1000),"kobs_",substr(Sys.time(),1,10),".csv")
  Mode       <- read.csv(NameRCT1)[,-1]
  Summ       <- read.csv(NameRCT2)[,-1]
  NameCon1   <- paste0("Mode_Con_Ignore_",Nreps,"reps_",
               (Nobs/1000),"kobs_",substr(Sys.time(),1,10),".csv")
  NameCon2   <- paste0("Summ_Con_Ignore_",Nreps,"reps_",
                    (Nobs/1000),"kobs_",substr(Sys.time(),1,10),".csv")
  Mode       <- rbind(Mode,read.csv(NameCon1)[,-1])
  Summ       <- rbind(Summ,read.csv(NameCon2)[,-1])
  NameMed1   <- paste0("Mode_Med_Ignore_",Nreps,"reps_",
                (Nobs/1000),"kobs_",substr(Sys.time(),1,10),".csv")
  NameMed2   <- paste0("Summ_Med_Ignore_",Nreps,"reps_",
```

```
                    (Nobs/1000),"kobs_",substr(Sys.time(),1,10),".csv")
  Mode       <- rbind(Mode,read.csv(NameMed1)[,-1])
  Summ       <- rbind(Summ,read.csv(NameMed2)[,-1])
  Name1      <- paste0("Mode_Ignore_",Nreps,"reps_",(Nobs/1000)
                              ,"kobs_",substr(Sys.time(),1,10),".csv")
  Name2      <- paste0("Summ_Ignore_",Nreps,"reps_",(Nobs/1000),
                           "kobs_",substr(Sys.time(),1,10),".csv")
  write.table(t(colnames(Mode)),Name1,sep=",",row.names=TRUE,
                       col.names=FALSE)
write.table(signif(Mode),Name1,sep=",",row.names=TRUE,
                           col.names=FALSE,append=TRUE)
  write.table(t(colnames(Summ)),Name2,sep=",",row.names=TRUE
                    ,col.names=FALSE)
  write.table(signif(Summ),Name2,sep=",",row.names=TRUE
                                          ,col.names=FALSE,append=TRUE)
}
```

Appendix C

Supplementary details for Chapter 5

C.1 Rcode

```
#Create Mplus text file for LCA and save it as mplus1.txt
[[init]]
iterators = classes;
classes = 1:7;
filename = "[[classes]]-classLCA.inp";
outputDirectory = "/Users/LCA";
[[/init]]

TITLE: Latent Class Analysis;
DATA: FILE = "Data.dat";
VARIABLE:
NAMES = PatientID MaleSex Diabetes StatusDeath TimeDeath
ClinicAge Haemoglobin;
USEVARIABLES = Diabetes MaleSex  ClinicAge Haemoglobin;
CATEGORICAL = Diabetes  MaleSex;
CLASSES = c ([[classes]]);
MISSING=.;
ANALYSIS: TYPE = MIXTURE;
SAVEDATA:
```

```
        FILE IS Data2.dat;
        SAVE IS cprob;
        FORMAT IS free;
#Create another Mplus text file for LCR and save it as mplus2.txt
[[init]]
iterators = classes;
classes = 2:5;
filename = "[[classes]]-classLCR.inp";
outputDirectory = "/Users/londt4/Documents/PhDwork/LCR";
[[/init]]

TITLE: Latent Class Analysis;
DATA: FILE = "Data.dat";

VARIABLE:
NAMES = PatientID MaleSex Diabetes StatusDeath TimeDeath
ClinicAge Haemoglobin Sex2;
 USEVARIABLES = Diabetes StatusDeath TimeDeath
 ClinicAge Haemoglobin Sex2;
    SURVIVAL = TimeDeath(ALL);
    TIMECENSORED = StatusDeath(1 = NOT 0 = RIGHT);


CLASSES = c ([[classes]]);
MISSING=.;
ANALYSIS:
ESTIMATOR = ML;
    TYPE = MIXTURE ;
    STARTS = 1000 20;
    PROCESSORS = 2;
MODEL:

    %OVERALL%
    TimeDeath on Diabetes ClinicAge Sex2;
    c on Diabetes Haemoglobin;
    %c#1%
    TimeDeath on Diabetes ClinicAge Sex2;
    %c#2%
    TimeDeath on Diabetes ClinicAge Sex2;
SAVEDATA:
        FILE IS Data2.dat;
        SAVE IS cprob;
        FORMAT IS free;
```

```
# Clear the working environment
rm(list=ls())
#Installing required packages
packages  <- c("foreign", "MplusAutomation", "dplyr", "tidyverse",
 "MASS", "ROCR", "tidyr","pROC","synthpop")
new_packages<-packages[!(packages %in%
installed.packages()[,"Package"])]
# Load all required packages
if(length(new_packages)) install.packages(new_packages)
for (i in 1:length(packages)) require(packages[i],character.only = T)

# import Data from SPSS
MyData  <- tibble(read.spss(file="heart.sav", to.data.frame = TRUE))
MyData  <- na.omit(MyData[ ,c(1,3,4,17,18,25,33)])
MyData$TimeDeath  <- MyData$TimeDeath / 365.2422
MyData$Sex2       <- as.integer(MyData$MaleSex)

################################################
####Latent Class Analysis Section ###################
################################################
prepareMplusData(MyData, file="Data.dat")
createModels("mplus1.txt")
runModels()
models=readModels()

################################################
####Latent Class Regression Section ##############
################################################
prepareMplusData(MyData, file="Data.dat")
createModels("mplus2.txt")
runModels()
summary=extractModelSummaries()
models=readModels()

###############
## Functions ##
###############

# Defining the function
cindex   <- function(data){
```

```
 # Define event time variable, status variable, and a
 risk score (A linear predictor)
  time     <- data$time
  status   <- data$status
  x        <- data$LP
  n        <- length(time)

   # Order variables on time (ascending), and
   # on status (descending - 1s first)
  # ord <- order(time,-status)
  ord      <- order(time)
  time     <- time[ord]
  status   <- status[ord]
  x        <- x[ord]

  #Select only individuals who experienced the event
  wh       <- which(status==1)

  # Every individual i with an event is compared to all other i
  ndividuals
  # with a later event time j with event times sorted in
  ascending order
  total    <- concordant <- 0
  for (i in wh) {
    for (j in ((i+1):n)) {
      if (time[j] > time[i]) { # ties not counted
        total <- total + 1
        # The total number of concordant and tied pairs is counted
        if (x[j] < x[i])  concordant <- concordant + 1
        if (x[j] == x[i]) concordant <- concordant + 0.5
      }
    }
  }

  # The proportion of concordant pairs over the total of
   evaluable pairs gives the C-index
  return(concordant/total)
}
```

```
# create input file for standard Cox PH (1-Class) model in Mplus
StdCox <- function(data) {
  CoxModel <- mplusObject(
    TITLE = "RealDataset;",
    VARIABLE = USEVARIABLES=TimeDeath StatusDeath
    Diabetes  MaleSex ClinicAge  Haemoglobin;
  SURVIVAL = TimeDeath(ALL);
  TIMECENSORED = StatusDeath(1 = NOT 0 = RIGHT);
  CLASSES=cl(1);,
    ANALYSIS = ESTIMATOR = ML;
  TYPE = MIXTURE ;
  STARTS = 1000 20;
  PROCESSORS = 8;,
    MODEL=
  %OVERALL%
  TimeDeath on Diabetes  ClinicAge MaleSex Haemoglobin;,
    SAVEDATA =
  FILE IS sim.dat;
  SAVE IS CPROBABILITIES;
  FORMAT IS free;,
    usevariables = c("MaleSex", "Diabetes", "StatusDeath", "TimeDeath",
     "ClinicAge", "Haemoglobin"), rdata = Data
  )
}

# create input file for Cox PH Latent Class Regression model in Mplus
LCRCox <- function(data)  {
  LCRCoxModel <- mplusObject(
    TITLE = "RealDataset;",
    VARIABLE = USEVARIABLES = TimeDeath StatusDeath Diabetes
     MaleSex ClinicAge  Haemoglobin;
  SURVIVAL = TimeDeath(ALL);
  TIMECENSORED = StatusDeath(1 = NOT 0 = RIGHT);
  CLASSES=cl(2);,
    ANALYSIS = ESTIMATOR = ML;
  TYPE = MIXTURE ;
  STARTS = 1000 20;
  PROCESSORS = 8;,
    MODEL=
  %OVERALL%
  TimeDeath on Diabetes  ClinicAge MaleSex;
  cl on Haemoglobin Diabetes;
  %cl#1%
```

```
  TimeDeath on Diabetes  ClinicAge MaleSex;
  %cl#2%
  TimeDeath on Diabetes  ClinicAge MaleSex;,
    SAVEDATA =
  FILE IS sim.dat;
  SAVE IS CPROBABILITIES;
  FORMAT IS free;,
    usevariables = c("MaleSex", "Diabetes", "StatusDeath", "TimeDeath",
     "ClinicAge", "Haemoglobin"), rdata = Data
  )
}

##############################
## Model Evaluation Section ##
##############################

# evaluate standard Cox PH model
StdMod  <- StdCox(Data)
StdFit  <- mplusModeler(StdMod, modelout = "StdData.inp", run = 1L)
StdData <- StdFit$results$savedata
StdOut  <- readModels("StdData.out", what = "all")
StdBeta <- as.numeric(StdOut$parameters$unstandardized[1:5,"est"])

LP      <- (StdData$DIABETES - mean(StdData$DIABETES)) * StdBeta[1]
+ (StdData$CLINICAG - mean(StdData$CLINICAG)) * StdBeta[2] +
  (StdData$MALESEX  - mean(StdData$MALESEX))  * StdBeta[3] +
   (StdData$HAEMOGLO - mean(StdData$HAEMOGLO)) * StdBeta[4]

# Calculating C-index for the standard Cox PH model
time    <- Data$TimeDeath
status  <- Data$StatusDeath
StdX    <- tibble(data.frame(time, status, LP))
C_Std   <- c_index(StdX)


# evaluate Cox PH LCR model
LCRMod  <- LCRCox(Data)
LCRFit  <- mplusModeler(LCRMod, modelout = "LCRData.inp", run = 1L)
LCRData <- LCRFit$results$savedata
LCROut  <- readModels("LCRData.out", what = "all")
LCRBeta <- as.numeric(LCROut$parameters$unstandardized[1:8,"est"])
```

```
# Linear predictor for class 1  plus linear predictor for
 class 2 weighted by class probabilities
LP      <- ( ((LCRData$DIABETES - mean(LCRData$DIABETES)) * LCRBeta[1]
+ (LCRData$CLINICAG - mean(LCRData$CLINICAG)) * LCRBeta[2] +
(LCRData$MALESEX  - mean(LCRData$MALESEX))  * LCRBeta[3] +
LCRBeta[4] ) * LCRData$CPROB1 ) +
 ( ((LCRData$DIABETES - mean(LCRData$DIABETES)) * LCRBeta[5] +
  (LCRData$CLINICAG - mean(LCRData$CLINICAG)) * LCRBeta[6] +
       (LCRData$MALESEX  - mean(LCRData$MALESEX))  * LCRBeta[7] +
        LCRBeta[8] ) * LCRData$CPROB2 )

#Evaluate the c-index for a Latent Class Cox Model
time    <- Data$TimeDeath
status  <- Data$StatusDeath
LCRX    <- tibble(data.frame(time, status, LP))
C_LCR   <- c_index(LCRX)

############################################################
######################CROSSVALIDATION   #################
############################################################
# 1. Partition Data into Training and Test Set
# 2. Create Syntax for training dataset
# 3. Run model #2
# 4. Take parameter estimates from #3 and create syntax
         #with those estimates as fixed values and test dataset
# 5. Run model #4
# 6. Output model fit information from #5 and save to a file
# 7. Repeat 10 times (10-fold CV)

set.seed(20210306)
# loopReplace function is needed to fill in parameters in
Mplus script with specified values

loopReplace <- function(text, replacements) {
  for (v in names(replacements)){
    text <- gsub(sprintf("\\[\\[%s\\]\\]",v),replacements[[v]],text)
  }
  return(text)
}
STDCoxCindex1 = matrix(NA,10,1)
STDCoxCindex2 = matrix(NA,10,1)
LCCoxCindex1 = matrix(NA,10,1)
LCCoxCindex2 = matrix(NA,10,1)
```

```
###### 10-fold Cross-Validation for the standard Cox Model #########
for(i in 1:10){
  #Segment data by fold using the which() function
  testIndexes <- which(folds==i,arr.ind=TRUE)
  testData <- MyData[testIndexes, ]
  trainData <- MyData[-testIndexes, ]

  # Create Syntax for the standard Cox Training Data
  STDtrain_script <- mplusObject(
    TITLE = Standard Cox Model using a Training Dataset;,
    VARIABLE = USEVARIABLES = Diabetes StatusDeath TimeDeath
                                  ClinicAge Haemoglobin Sex2;
    SURVIVAL = TimeDeath(ALL);
    TIMECENSORED = StatusDeath(1 = NOT 0 = RIGHT);
    CLASSES=c(1);,
    ANALYSIS = ESTIMATOR = ML;
    TYPE = MIXTURE ;
    STARTS = 1000 20;
    PROCESSORS = 2;,
    MODEL=
    %OVERALL%
    TimeDeath on Diabetes ClinicAge Sex2 Haemoglobin;,
    SAVEDATA =
        FILE IS STDtrainData.dat;
        SAVE IS cprob;
        FORMAT IS free;,
    usevariables = c( "Diabetes"  ,"StatusDeath" ,"TimeDeath",
"ClinicAge" , "Haemoglobin","Sex2" ),rdata = trainData)

  STDtrainModel = mplusModeler(STDtrain_script,
  modelout = "STDtrainModel.inp", run = 1L)
  STDTrainData <- STDtrainModel$results$savedata
  STDOut  <- readModels("STDtrainModel.out", what = "all")
  STDBeta <- as.numeric(STDOut$parameters$unstandardized[1:5,"est"])

  #Linear predictor for the standard Cox model using the training data
  LP <-(STDTrainData$DIABETES - mean(STDTrainData$DIABETES))*STDBeta[1]
  + (STDTrainData$CLINICAG - mean(STDTrainData$CLINICAG)) * STDBeta[2]+
   (STDTrainData$SEX2 - mean(STDTrainData$SEX2))*STDBeta[3] +
   (STDTrainData$HAEMOGLO-mean(STDTrainData$HAEMOGLO))*STDBeta[4]

  #Evaluate the c-index for a Stadard Cox Model using training data
  time    <- trainData$TimeDeath
```

```
  status  <- trainData$StatusDeath
  STDX    <- tibble(data.frame(time, status, LP))
  STDCoxCindex1[i,1]<- c_index(STDX)

  STDparms = STDtrainModel$results$parameters
  df_parms <- data.frame(STDparms[[1]]$param, STDparms[[1]]$est)
  names(df_parms) <- c("param","est")
  df_parms2 <- t(df_parms)
  df_parms2 <- as.data.frame(df_parms2)
  df2 <- df_parms2[2,]
  names(df2) <- c("DIAB_c1","AGE_c1","SEX_c1","HAEM_c1","TIME_c1")

  # Create Syntax for the standard Cox model for the Test Data
  STDtest_script <- mplusObject(
    TITLE = Standard Cox Model on a Testing dataset;,
    VARIABLE = USEVARIABLES = Diabetes
    StatusDeath TimeDeath ClinicAge
            Haemoglobin Sex2;
  SURVIVAL = TimeDeath (ALL);
  TIMECENSORED = StatusDeath(1 = NOT 0 = RIGHT);
  CLASSES=c(1);,
    ANALYSIS = TYPE=MIXTURE,
    MODEL = loopReplace(
                      %OVERALL%
                      TimeDeath on Diabetes ClinicAge Sex2 Haemoglobin;
                      TimeDeath on Diabetes@[[DIAB_c1]]
                      ClinicAge@[[AGE_c1]]
                      Sex2@[[SEX_c1]]
                      Haemoglobin@[[HAEM_c1]];
                      [TimeDeath@[[TIME_c1]]];
                      OUTPUT: NOCHISQUARE;, df2),
    SAVEDATA =
      FILE IS STDtestData.dat;
      SAVE IS cprob;
      FORMAT IS free;,
    usevariables = c("Diabetes"  ,"StatusDeath" ,"TimeDeath",
"ClinicAge" , "Haemoglobin","Sex2" ), rdata = testData)
  STDtest = mplusModeler(STDtest_script,
  modelout = "STDtestModel.inp", run = 1L)
  STDTestData <-STDtest$results$savedata
  head(STDTestData)
  STDTestOut  <- readModels("STDtestModel.out", what = "all")
  STDTestBeta <-
```

```
as.numeric(STDTestOut$parameters$unstandardized[1:8,"est"])


  # Linear predictor for the Standard Cox Model using the testing data
  LP <-(STDTestData$DIABETES-mean(STDTestData$DIABETES))*STDTestBeta[1]
  +(STDTestData$CLINICAG-mean(STDTestData$CLINICAG)) * STDTestBeta[2]
  +(STDTestData$SEX2 - mean(STDTestData$SEX2))*STDTestBeta[3] +
    (STDTestData$HAEMOGLO-mean(STDTestData$HAEMOGLO))*STDTestBeta[4]

  #Evaluate the c-index for a Standard Cox Model using testing data
  time     <- testData$TimeDeath
  status   <- testData$StatusDeath
  STDX     <- tibble(data.frame(time, status, LP))
  STDCoxCindex2[i,1]<- c_index(STDX)

  # Create Syntax for the Latent Class Cox regression Training Data
  LCRtrain_script <- mplusObject(
    TITLE = Latent Class Regression on Training dataset;,
    VARIABLE = USEVARIABLES = Diabetes StatusDeath TimeDeath
     ClinicAge Haemoglobin Sex2;
    SURVIVAL = TimeDeath(ALL);
    TIMECENSORED = StatusDeath(1 = NOT 0 = RIGHT);
    CLASSES=c(2);,
    ANALYSIS = ESTIMATOR = ML;
    TYPE = MIXTURE ;
    STARTS = 1000 100;
    PROCESSORS = 4;,
    MODEL=
    %OVERALL%
    TimeDeath on Diabetes ClinicAge Sex2;
    c on Diabetes Haemoglobin;
    %c#1%
    TimeDeath on  Diabetes ClinicAge Sex2;
    %c#2%
    TimeDeath on  Diabetes ClinicAge Sex2;,
    SAVEDATA =
        FILE IS LCRtrainData.dat;
        SAVE IS cprob;
        FORMAT IS free;,
    usevariables = c( "Diabetes"  ,"StatusDeath" ,"TimeDeath",
"ClinicAge" , "Haemoglobin","Sex2" ),rdata = trainData)
  LCRtrain = mplusModeler(LCRtrain_script,
```

```
                        modelout = "LCRtrainModel.inp", run = 1L)
LCRtrainData <-LCRtrain$results$savedata
LCRtrainOut  <- readModels("LCRtrainModel.out", what = "all")
LCRtrainBeta<- as.numeric(LCRtrainOut$parameters$unstandardized
                                  [1:8,"est"])
# Linear predictor for class 1 plus linear predictor
  #    for class 2 weighted by class probabilities
LP <-
(((LCRtrainData$DIABETES-mean(LCRtrainData$DIABETES))*LCRtrainBeta[1]
+ (LCRtrainData$CLINICAG-mean(LCRtrainData$CLINICAG))*LCRtrainBeta[2]
+(LCRtrainData$SEX2-mean(LCRtrainData$SEX2))*LCRtrainBeta[3]
+ LCRtrainBeta[4] ) * LCRtrainData$CPROB1 ) +
(((LCRtrainData$DIABETES-mean(LCRtrainData$DIABETES))*LCRtrainBeta[5]
+ (LCRtrainData$CLINICAG-mean(LCRtrainData$CLINICAG))*LCRtrainBeta[6]
+ (LCRtrainData$SEX2-mean(LCRtrainData$SEX2))*LCRtrainBeta[7]
+ LCRtrainBeta[8])*LCRtrainData$CPROB2 )

#Evaluate the c-index for a Latent Class Cox Model
  #    using a testing dataset
time     <- LCRtrainData$TIMEDEAT
status   <- trainData$StatusDeath
LCRX     <- tibble(data.frame(time, status, LP))
LCCoxCindex1[i,1] = c_index(LCRX)

parms = LCRtrain$results$parameters
df_parms <- data.frame(parms[[1]]$param, parms[[1]]$est)
names(df_parms) <- c("param","est")
df_parms2 <- t(df_parms)
df_parms2 <- as.data.frame(df_parms2)
df2 <- df_parms2[2,]
names(df2) <- c("DIAB_c1","AGE_c1","SEX_c1","TIME_c1","DIAB_c2",
"AGE_c2","SEX_c2","TIME_c2","DIAB_cc1","HAEM_cc1","c1_cc1")

# Create Syntax for the LCR Test Data
LCRtest_script <- mplusObject(
  TITLE = Latent Class Regression on Testing dataset;,
  VARIABLE = USEVARIABLES = Diabetes
  StatusDeath TimeDeath
   ClinicAge Haemoglobin Sex2;
SURVIVAL = TimeDeath (ALL);
TIMECENSORED = StatusDeath(1 = NOT 0 = RIGHT);
CLASSES=c(2);,
  ANALYSIS = TYPE=MIXTURE,
```

```
  MODEL = loopReplace(
                    %OVERALL%
                    TimeDeath on Diabetes ClinicAge Sex2;
                    c#1 on Diabetes Haemoglobin;
                    c#1 on Diabetes@[[DIAB_cc1]]
                    Haemoglobin@[[HAEM_cc1]];
                    %c#1%
                    TimeDeath on Diabetes@[[DIAB_c1]]
                    ClinicAge@[[AGE_c1]]
                    Sex2@[[SEX_c1]];
                    [TimeDeath@[[TIME_c1]]];
                    %c#2%
                    TimeDeath on Diabetes@[[DIAB_c2]]
                    ClinicAge@[[AGE_c2]]
                    Sex2@[[SEX_c2]];

                    [TimeDeath@[[TIME_c2]]];

                    OUTPUT: NOCHISQUARE;, df2),
  SAVEDATA =
    FILE IS LCRtestData.dat;
    SAVE IS cprob;
    FORMAT IS free;,
  usevariables = c( "Diabetes"  ,"StatusDeath" ,"TimeDeath",
        "ClinicAge" , "Haemoglobin","Sex2" ), rdata = testData)
LCRtest = mplusModeler(LCRtest_script,modelout = "LCRtestModel.inp",
                  run = 1L)
LCRtestData <-LCRtest$results$savedata
LCRtestOut  <- readModels("LCRtestModel.out", what = "all")
LCRtestBeta <-
 as.numeric(LCRtestOut$parameters$unstandardized[1:8,"est"])

# Linear predictor for class 1  plus linear predictor for
 #            class 2 weighted by class probabilities
LP <-
(((LCRtestData$DIABETES-mean(LCRtestData$DIABETES))*LCRtestBeta[1]
+ (LCRtestData$CLINICAG-mean(LCRtestData$CLINICAG))*LCRtestBeta[2]
+ (LCRtestData$SEX2-mean(LCRtestData$SEX2))*LCRtestBeta[3]
+ LCRtestBeta[4] )*LCRtestData$CPROB1 )
+(((LCRtestData$DIABETES-mean(LCRtestData$DIABETES))*LCRtestBeta[5]
+ (LCRtestData$CLINICAG-mean(LCRtestData$CLINICAG))*LCRtestBeta[6]
+(LCRtestData$SEX2-mean(LCRtestData$SEX2))*LCRtestBeta[7]
+ LCRtestBeta[8])*LCRtestData$CPROB2)
```

```
  #Evaluate the c-index for a Latent Class Cox Model using
  #       a testing dataset
  time     <- LCRtestData$TIMEDEAT
  status   <- testData$StatusDeath
  LCRX     <- tibble(data.frame(time, status, LP))
  LCCoxCindex2[i,1]    <- c_index(LCRX)
}

# Summary statistics(minimum, lower-hinge, median,upper-hinge, maximum)
fivenum(STDCoxCindex1)
fivenum(STDCoxCindex2)
fivenum(LCCoxCindex1)
fivenum(LCCoxCindex2)
```

Appendix D

Supplementary details for Chapter 6

D.1 Rcode for simulations

```
# Clear workspace
rm(list=ls())

#Installing required packages
packages<- c("survival", "gridExtra", "matrixStats", "dplyr",
                    "tidyverse", "dagitty","MASS", "rpsychi",
                    "ROCR","MplusAutomation", "pROC", "matrixcalc",
                     "summarytools", "pec", "riskRegression")

new_packages  <- packages[!(packages %in%
                                  installed.packages()[,"Package"])]

if(length(new_packages)) install.packages(new_packages)
for (i in 1:length(packages))
require(packages[i],character.only = T)
```

```
##################
## setup values ##
##################

op          <- options(digits.secs = 6)
Nobs        <- 1000 # study sample size
reps        <- 1000 # number of bootstrap simulations/replications
names0      <- c("S0",  "C0", "X1", "X2", "X3")
names1      <- c("S1",  "C1", "X1", "X2", "X3", "DTH")
names2      <- c("Bin", "C1", "X1", "X2", "X3")

#######################################################
## assign DAG path coefficients for each scenario ##
#######################################################

DAG <- function(x) {
  if (x == 1) {
    # DAG1a
    # S0/S1:
    #heart failure among patients with coronary hearth disease
    # x1: SEB - early-life socioeconomic background influences
                  #  lifestyle and mediating factors that lead
                   # population heterogeneity
    # X2: SMK - smoking history
    # X3: ADH - drug adherence
    # C0/C1: mix of lifestyle, diet, exercise and
           # other multifactortial issues

    X1X2 <<- 0.25    # modest early-life SEB
                # influence on smoking behaviours
                 ## THIS CHANGES ##
    X1X3 <<- 0.25    # modest early-life SEB influences on
     #  later-life treatment adherence
     ## THIS CHANGES ##
    X2X3 <<- 0.10    # modest link between smoking
    #  behaviours and treatment adherence behaviour
    X1C0 <<- 0.50    # strong early-life SEB influences on
     # latent population heterogeneity
    C0S0 <<- 0.50    # strong latent population heterogeneity
         # influence on outcome; NB:
         # indirect X1S0 effect is modest (0.25)
    X2S0 <<- 0.50    # strong influence of smoking on outcome
```

```
    X2C0 <<- 0.10     # modest influence of smoking behaviours on
          # latent population heterogeneity that is not
                  # explained by SEB
    C0X3 <<- 0.05    # weak influence of latent population
              # heterogeneity on drug adherence
    X3S0 <<- 0.50     # strong influence of drug adherence on outcome
    X1S0 <<- 0.00     # zero influence of early-life SEB on outcome not
          # mediated through latent population heterogeneity

           ## THIS CHANGES ##
  }
  if (x == 2) {
    # DAG1b
    # X1: GEN - family history suggestive of underlying genetic
            #  predisposition to both CVD and early death

    X1X2 <<- 0.05      # weak genetic predisposition to
                              #  smoking behaviours
                              ## THIS CHANGES ##
    X1X3 <<- 0.20     # modest genetic influences on
                               # treatment adherence
                                ## THIS CHANGES ##
    X2X3 <<- 0.10      # modest influence of smoking behaviours on
                                # treatment adherence behaviour
    X1C0 <<- 0.50  # strong genetic influences on latent population
                   # heterogeneity
    C0S0 <<- 0.50    # strong latent population heterogeneity
                            #  influence on outcome; NB: indirect
                              # X1S0 effect is modest (0.25)
    X2S0 <<- 0.50        # strong influence of smoking on outcome
    X2C0 <<- 0.10        # modest influence of smoking behaviours
                                # on latent population heterogeneity
                                # that is
                                # not explained by genetics
    C0X3 <<- 0.10        # modest influence of latent
                                # population heterogeneity on
                                # drug adherence
    X3S0 <<- 0.50        # strong influence of drug adherence on outcome
    X1S0 <<- 0.20        # modest-to-strong genetic
                               # influence on outcome not
            # explained by latent population heterogeneity
## THIS CHANGES ##
  }
```

```
if (x == 3) {
  # DAG1b
  # X1: GEN - family history suggestive of underlying genetic
 #  predisposition to both CVD and early death

  X1X2 <<- 0.05        # weak genetic predisposition to
                              # smoking behaviours
  X1X3 <<- 0.20        # modest genetic influences on
                              # treatment adherence
  X2X3 <<- 0.10        # modest influence of smoking behaviours on
                              #  treatment adherence behaviour
  X1C0 <<- 0.00        # NO # genetic influences on latent
                              #  population heterogeneity
         ## THIS CHANGES ##
  C0S0 <<- 0.50        # strong latent population heterogeneity
                             #   influence on outcome;
                             # NB: indirect X1S0 effect
                             # is modest (0.25)
  X2S0 <<- 0.50        # strong influence of smoking on outcome
  X2C0 <<- 0.00        # NO # influence of smoking
                              # on latent population
                            # heterogeneity that is
                            # not explained by genetics

                              ## THIS CHANGES ##
  C0X3 <<- 0.00        # NO # influence of latent population
                           # heterogeneity on drug adherence
         ## THIS CHANGES ##
  X3S0 <<- 0.50        # strong influence of drug adherence on
                              #  outcome
  X1S0 <<- 0.50        # strong genetic influence on outcome
     # not explained by latent population heterogeneity
}
return(x) }
```

```
#####################
## other functions ##
#####################

# create DAG
MakeDAG <- function(x) {
  DAG(x)
  dag   <- paste0(dag {
   X1 -> X2 [beta = , X1X2, ]
   X1 -> X3 [beta = , X1X3, ]
   X2 -> X3 [beta = , X2X3, ]
   X1 -> C0 [beta = ", X1C0, "]
   C0 -> S0 [beta = ", C0S0, "]
   X2 -> S0 [beta = ", X2S0, "]
   X2 -> C0 [beta = ", X2C0, "]
   C0 -> X3 [beta = ", C0X3, "]
   X3 -> S0 [beta = ", X3S0, "]
   X1 -> S0 [beta = ", X1S0, "] }
  )
  return(dag) }

# create input file for Mplus with modest number of starts
LCRSurv <- function(data)  {
  LCR   <- mplusObject(
    TITLE = "Simulations;",
    VARIABLE = "USEVARIABLES = X1 X2 X3 DTH S1;
     SURVIVAL = S1(ALL);
     TIMECENSORED = DTH(1 = NOT 0 = RIGHT);
     CLASSES = CL(2);",
    ANALYSIS = "ESTIMATOR = ML;
     TYPE = MIXTURE ;
     STARTS = 200 20;
     PROCESSORS = 8;",
    MODEL="
     %OVERALL%
     S1 on X1 X2 X3;
     CL on X1 X2;
     %CL#1%
     S1 on X1 X2 X3;
     %CL#2%
     S1 on X1 X2 X3;",
    SAVEDATA = "
     FILE IS sim.dat;
```

```
     SAVE IS CPROBABILITIES;
     FORMAT IS free;",
    usevariables = c("S1", "DTH", "X1", "X2", "X3"), rdata = data)}

# create input file for Mplus with more starts if needed
LCRSurvExtra <- function(data)  {
  LCR   <- mplusObject(
    TITLE = "Simulations;",
    VARIABLE = "USEVARIABLES = X1 X2 X3 DTH S1;
     SURVIVAL = S1(ALL);
     TIMECENSORED = DTH(1 = NOT 0 = RIGHT);
     CLASSES = CL(2);",
    ANALYSIS = "ESTIMATOR = ML;
     TYPE = MIXTURE ;
     STARTS = 2000 200;
     PROCESSORS = 8;",
    MODEL="
     %OVERALL%
     S1 on X1 X2 X3;
     CL on X1 X2;
     %CL#1%
     S1 on X1 X2 X3;
     %CL#2%
     S1 on X1 X2 X3;",
    SAVEDATA = "
     FILE IS sim.dat;
     SAVE IS CPROBABILITIES;
     FORMAT IS free;",
    usevariables = c("S1", "DTH", "X1", "X2", "X3"), rdata = data)
}

# create input file for Mplus with modest number of starts
LCRSurvNoX1 <- function(data)  {
  LCR   <- mplusObject(
    TITLE = "Simulations;",
    VARIABLE = "USEVARIABLES = X2 X3 DTH S1;
     SURVIVAL = S1(ALL);
     TIMECENSORED = DTH(1 = NOT 0 = RIGHT);
     CLASSES = CL(2);",
    ANALYSIS = "ESTIMATOR = ML;
     TYPE = MIXTURE ;
     STARTS = 500 50;
     PROCESSORS = 8;",
```

```
    MODEL="
     %OVERALL%
     S1 on X2 X3;
     CL on X2;
     %CL#1%
     S1 on X2 X3;
     %CL#2%
     S1 on X2 X3;",
    SAVEDATA = "
     FILE IS sim.dat;
     SAVE IS CPROBABILITIES;
     FORMAT IS free;",
    usevariables = c("S1", "DTH", "X2", "X3"), rdata = data)}

# create input file for Mplus with more starts if needed
LCRSurvExtraNoX1 <- function(data)  {
  LCR   <- mplusObject(
    TITLE = "Simulations;",
    VARIABLE = "USEVARIABLES = X2 X3 DTH S1;
     SURVIVAL = S1(ALL);
     TIMECENSORED = DTH(1 = NOT 0 = RIGHT);
     CLASSES = CL(2);",
    ANALYSIS = "ESTIMATOR = ML;
     TYPE = MIXTURE ;
     STARTS = 2000 200;
     PROCESSORS = 8;",
    MODEL="
     %OVERALL%
     S1 on X2 X3;
     CL on X2;
     %CL#1%
     S1 on X2 X3;
     %CL#2%
     S1 on X2 X3;",
    SAVEDATA = "
     FILE IS sim.dat;
     SAVE IS CPROBABILITIES;
     FORMAT IS free;",
    usevariables = c("S1", "DTH", "X2", "X3"), rdata = data) }
```

```
# create input file for Mplus with more starts if needed
LCRSurvExtraExtraNoX1 <- function(data)  {
  LCR   <- mplusObject(
    TITLE = "Simulations;",
    VARIABLE = "USEVARIABLES = X2 X3 DTH S1;
     SURVIVAL = S1(ALL);
     TIMECENSORED = DTH(1 = NOT 0 = RIGHT);
     CLASSES = CL(2);",
    ANALYSIS = "ESTIMATOR = ML;
     TYPE = MIXTURE ;
     STARTS = 5000 500;
     PROCESSORS = 8;",
    MODEL="
     %OVERALL%
     S1 on X2 X3;
     CL on X2;
     %CL#1%
     S1 on X2 X3;
     %CL#2%
     S1 on X2 X3;",
    SAVEDATA = "
     FILE IS sim.dat;
     SAVE IS CPROBABILITIES;
     FORMAT IS free;",
    usevariables = c("S1", "DTH", "X2", "X3"), rdata = data) }

# derive S1 such that all subjects die within 20
# years and we have $70\%$ $5$-year survival -
 # censoring at $5$ years only
GetS1   <- function(S0) {
  span  <- 25
  gap   <- 0
  Delta <- 1
  while (abs(Delta) > 0.2) {
    Optimal          <-  optimise(
      Exponentiate   <<- function(x) {
        S1           <<- rexp(Nobs,x)
        Extra        <-  runif(length(S1[S1>span]), gap, span)
        S1[S1>span] <<- Extra
        return(5 - quantile(S1,0.3))
      }, c(0.05,0.5)
    )
    Delta  <- Optimal$objective
```

```
  }
  Map <-  bind_cols(id = 1:Nobs, S0 = S0)
  Map <-  bind_cols(Map[order(S0),], S1 = S1[order(S1)])
  S1  <<- Map[order(Map$id),]$S1
  return(Optimal$minimum) }

# Derive Harrell's C-statistic from Cox PH model and given data set
C_index     <- function(ph, data) {
  Xstart    <- 1
  if (class(ph) == "coxph") {
    beta    <- ph$coefficients
    nCovars <- length(beta)
  } else {
    beta    <- ph
    Xstart  <- if_else (length(beta)/4 == round(length(beta)/4), 1, 2)
    nCovars <- length(grep("X",names(data)))
  }
  nClass    <- length(grep("CPROB",names(data)))
  time      <- data$S1
  status    <- data$DTH
  if (nClass==0) LP <- as.matrix(data[,paste0("X",Xstart:nCovars)],
          ncol = nCovars) %*% beta else
    if (nClass==2) {
      Cprob <- data[,c("CPROB1","CPROB2")]
      LPC1  <- (as.matrix(data[,paste0("X",Xstart:nCovars)],
          ncol = nCovars) %*% beta[1:nCovars])
* Cprob[,1]
      LPC2  <- (as.matrix(data[,paste0("X",Xstart:nCovars)],

          ncol = nCovars) %*% beta[(nCovars+1)
                :(2*nCovars)]) * Cprob[,2]

LP    <- as.numeric(unlist(LPC1 + LPC2)) }
  n         <- length(time)
  ord       <- order(time)
  time      <- time[ord]
  status    <- status[ord]
  LP        <- LP[ord]
  wh        <- which(status==1)
  total     <- concordant <- 0
  for (i in wh) {
    for (j in ((i+1):n)) {
      if (time[j] > time[i]) { # ties not counted
```

```
        total <- total + 1
        # The total number of concordant an tied pairs is counted
        if (LP[j] <  LP[i]) concordant <- concordant + 1
        if (LP[j] == LP[i]) concordant <- concordant + 0.5
      }
    }
  }
  # The proportion of concordant pairs over
   #  the total of evaluable pairs gives the C-index
  return(concordant/total) }

# calculate how well the LCR model performs
   # better than the standard model
PctBetter <- function(Sdata) {
  Better1 <- sum(Sdata$LCR1 >= Sdata$CoxC1, na.rm = TRUE)
  Worse1  <- sum(Sdata$LCR1 <  Sdata$CoxC1, na.rm = TRUE)
  Better2 <- sum(Sdata$LCR2 >= Sdata$CoxC2, na.rm = TRUE)
  Worse2  <- sum(Sdata$LCR2 <  Sdata$CoxC2, na.rm = TRUE)
  pct1    <- 100 * Better1 / (Better1 + Worse1)
  pct2    <- 100 * Better2 / (Better2 + Worse2)
  return(round(c(pct1, pct2),1)) }

################################################
## preliminary simulations to generate important
# summary descriptive information ##
################################################

# simulate empirical = TRUE data sets
dag               <- dagData <- Mu <- MyCov <- MyCor <- CorCov <-
                                    vector(mode = "list", length = 2)
set.seed(17)
for (itn in 1:3) {
  dag[[itn]]      <- MakeDAG(itn)
  dagData[[itn]]  <- simulateSEM(dag[[itn]], N = Nobs, eps = 1,
        standardized = FALSE, empirical = TRUE)[,names0] }

# derive correlation & covariance structures from empirical = TRUE
                                          simulated survival data
MedSurv <- vector(mode = "numeric", length = 2)
for (itn in 1:3) {
  data                <- tibble(dagData[[itn]])[,names0]
  Mu[[itn]]           <- as.numeric(round(data %>%
            summarise_if(is.numeric, mean),3))
```

```
  MyCov[[itn]]         <- var(dagData[[itn]])
  MyCor[[itn]]         <- cor(dagData[[itn]])
  dagData[[itn]]$C1    <- ifelse(data$C0 <= quantile(data$C0, 0.7),
                          0, 1)  # make binary latent class
  S1                   <- data$S0;
  x <- GetS1(dagData[[itn]]$S0)           # create exponential
       # survival data (global assignment to S1)
  dagData[[itn]]$S1    <- S1
CorCov[[itn]] <- lower.triangle('diag<-'(round(cor(dagData[[itn]]),2),
                 0)) + upper.triangle(round(var(dagData[[itn]]),2))
  MedSurv[itn]   <- median(dagData[[itn]]$S1)}

#############################################################
## main simulations of data sets to be evaluated by both models ##
#############################################################

# simulated all empirical = FALSE data sets from DAGs and store
MyData       <- vector(mode = "list", length = 6)
MyData[[1]] <- vector(mode = "list", length = reps)
t0           <- Sys.time()
for (itn in 1:3) {
  for(repi in 1:reps){
    set.seed(repi*Nobs)
    MyData[[itn]][[repi]]        <- tibble(data.frame(mvrnorm(Nobs,
    Mu[[itn]], MyCov[[itn]], empirical = FALSE))[,names0])
    MyData[[itn]][[repi]]$C1 <- ifelse(MyData[[itn]][[repi]]$C0 <=
    quantile(MyData[[itn]][[repi]]$C0, 0.7), 0, 1)  # make binary
                                                    #  latent class
    S1       <- MyData[[itn]][[repi]]$S0      # create survival data
    x       <- GetS1(MyData[[itn]][[repi]]$S0)   # exponentiate
    MyData[[itn]][[repi]]$S1    <- S1
    MyData[[itn]][[repi]]$DTH   <- ifelse(MyData[[itn]][[repi]]$S1
                                          < 5, 1, 0)
# 30\% binary death
    MyData[[itn]][[repi]][MyData[[itn]][[repi]]$S1 > 5,]$S1 <- 5 } }
# censor at 5 years
save(MyData, file = paste0(path,"MyData.rda"))
print("Data generation runtime"); print(Sys.time()-t0)
```

```
#####################################################
## model evaluation for all covariates selected  ##
#####################################################

t0  <- Sys.time()
for (itn in 1:3) {
  ts       <- Sys.time()
  CoxC1    <- CoxC2 <- LCR1  <- LCR2 <- Extra1
   <- Extra2 <- Fail1 <- Fail2 <- matrix(NA, reps, 1)
  for(repi in 1:reps){
    print("############################");
    print(paste0("# SCENARIO ",itn," REPLICATION ",repi));
    print("############################")
    Extra1[repi]         <- Extra2[repi] <- Fail1[repi]
                                <- Fail2[repi] <- 0
    ConData              <- tibble(MyData[[itn]][[repi]][,names1])
     # assign continuous predictors
    BinData              <- ConData
    # assign binary predictors
    BinData[,3:5]        <- apply(ConData[,3:5], 2, function(x){
     ifelse(x<=quantile(x,0.7), 0, 1) })
                                # convert X1, X2 & X3 to binary
    Cox1                 <- coxph(Surv(S1, C1) ~ X1 + X2 + X3,
    data = ConData, x = TRUE, y = TRUE)
    Cox2                 <- coxph(Surv(S1, C1) ~ X1 + X2 + X3,
    data = BinData, x = TRUE, y = TRUE)
    C1                   <- C_index(Cox1, ConData)
    C2                   <- C_index(Cox2, BinData)
    CoxC1[repi]          <- if_else(C1 >= 0.5, C1, 1 - C1)
    CoxC2[repi]          <- if_else(C2 >= 0.5, C2, 1 - C2)
    LCR1mod              <- LCRSurv(ConData[,names1])
    LCR2mod              <- LCRSurv(BinData[,names1])
    Fit1                 <- mplusModeler(LCR1mod, modelout =
    paste0("LCR1_S",itn,"_",repi,".inp"), run = 1L)
    Fit2                 <- mplusModeler(LCR2mod, modelout =
     paste0("LCR2_S",itn,"_",repi,".inp"), run = 1L)
    Out1  <- readModels(paste0("lcr1_s",itn,"_",repi,".out"),
                                          what = "all")
    Out2  <- readModels(paste0("lcr2_s",itn,"_",repi,".out"),
                                             what = "all")
    Err1  <- sum(grep("NON-POSITIVE", Out1$errors)) +
     sum(grep("DID NOT TERMINATE", Out1$errors))
    Err2                 <- sum(grep("NON-POSITIVE", Out2$errors))
```

```
                + sum(grep("DID NOT TERMINATE", Out2$errors))
War1                <- length(Out1$warnings)
War2                <- length(Out2$warnings)
X                   <- as.matrix(MyData[[itn]][[repi]][,
                              c("X1","X2","X3")])
if (Err1 == 0) {
  if (War1 != 0) {
    if (length(grep("BEST LOGLIKELIHOOD VALUE WAS NOT REPLICATED",
                              War1))!=0) {
      LCR1mod       <- LCRSurvExtra(BinData[,names1])
      Fit1          <- mplusModeler(LCR1mod, modelout =
                       paste0("LCR1_S",itn,"_",repi,".inp"),
                              run = 1L)
      Out1          <- readModels(paste0("lcr1_s",itn,"_",
                              repi,".out"), what = "all")
      War1          <- length(Out1$warnings)
      Extra1[repi]  <- 1
      Fail1[repi]   <- if_else (War1 == 0, 0, 1)
    }
  }
  if (War1 == 0) {
  Beta1   <- as.numeric(Out1$parameters$unstandardized[1:8,"est"])
  Check1  <- as.numeric(Out1$parameters$unstandardized[1:8,"se"])
    Beta1[Check1==0]<- 0
    if (sum(is.na(Beta1))==0 & length(Check1)!=0) {
      LCR1data      <- tibble(Fit1$results$savedata)
      ###################################
      LCR1data$DTH  <- 1 - LCR1data$DTH
      ###################################
      C1            <- C_index(Beta1, LCR1data)
      LCR1[repi]    <- if_else(C1 >= 0.5, C1, 1 - C1)
    } else LCR1[repi] <- NA
  }
} else LCR1[repi]   <- NA
if (Err2 == 0) {
  if (War2 != 0) {
    if (length(grep("BEST LOGLIKELIHOOD VALUE WAS
     NOT REPLICATED", War2))!=0) {
      LCR2mod       <- LCRSurvExtra(BinData[,names1])
      Fit2          <- mplusModeler(LCR2mod, modelout =
            paste0("LCR2_S",itn,"_",repi,".inp"), run = 1L)
      Out2          <- readModels(paste0("lcr2_s",itn,"_",repi,".
                              out"), what = "all")
```

```
          War2           <- length(Out2$warnings)
          Extra2[repi]   <- 1
          Fail2[repi]    <- if_else (War2 == 0, 0, 1)
        }
      }
      if (War2 == 0) {
        Beta2 <- as.numeric(Out2$parameters$unstandardized[1:8,"est"])
        Check2 <- as.numeric(Out2$parameters$unstandardized[1:8,"se"])
        Beta2[Check2==0]<- 0
        if (sum(is.na(Beta2))==0 & length(Check2)!=0) {
          LCR2data       <- tibble(Fit2$results$savedata)
          ###################################
          LCR2data$DTH   <- 1 - LCR2data$DTH
          ###################################
          C2             <- C_index(Beta2, LCR2data)
          LCR2[repi]     <- if_else(C2 >= 0.5, C2, 1 - C2)
        } else LCR2[repi] <- NA
      }
    } else LCR2[repi]    <- NA
  }
  Update                 <- data.frame(Scenario=itn,Model=1:reps,
                             CoxC1=CoxC1,LCR1=LCR1,
          Extra1=Extra1,Fail1=Fail1,CoxC2=CoxC2,
            LCR2=LCR2,Extra2=Extra2,Fail2=Fail2)
  Results     <- bind_rows(Results, Update)
}; print("Total runtime"); print(Sys.time()-t0)

#########################################################
## model evaluation for X1 covariate NOT selected  ##
#########################################################

t0  <- Sys.time()
for (itn in 1:3) {
  ts       <- Sys.time()
  CoxC1    <- CoxC2 <- LCR1  <- LCR2 <- Extra1 <-
                  Extra2 <- Fail1 <- Fail2 <- matrix(NA, reps, 1)
  for(repi in 1:reps){
    print("#############################");
     print(paste0("# SCENARIO ",itn + 3," REPLICATION
      ",repi));print("#############################")
    Extra1[repi]         <- Extra2[repi] <- Fail1[repi] <-
                              Fail2[repi] <- 0
    ConData              <- tibble(MyData[[itn]][[repi]][,names1])
```

```
                # assign continuous predictors
BinData                 <- ConData
                 # assign binary predictors
BinData[,3:5]           <- apply(ConData[,3:5], 2,
   function(x){ ifelse(x<=quantile(x,0.7), 0, 1) })
           # convert X1, X2 & X3 to binary
Cox1                    <- coxph(Surv(S1, C1) ~ X2 + X3,
                  data = ConData, x = TRUE, y = TRUE)
Cox2                    <- coxph(Surv(S1, C1) ~ X2 + X3,
                     data = BinData, x = TRUE, y = TRUE)
C1                      <- C_index(Cox1, ConData)
C2                      <- C_index(Cox2, BinData)
CoxC1[repi]             <- if_else(C1 >= 0.5, C1, 1 - C1)
CoxC2[repi]             <- if_else(C2 >= 0.5, C2, 1 - C2)
LCR1mod                 <- LCRSurvNoX1(ConData[,names1])
LCR2mod                 <- LCRSurvNoX1(BinData[,names1])
Fit1    <- mplusModeler(LCR1mod, modelout =
          paste0("LCR1_S",itn,"_",repi,".inp"), run = 1L)
Fit2                    <- mplusModeler(LCR2mod, modelout =
         paste0("LCR2_S",itn,"_",repi,".inp"), run = 1L)
Out1                    <- readModels(paste0("lcr1_s",itn,"_"
                                  ,repi,".out"), what = "all")
Out2                    <- readModels(paste0("lcr2_s",itn,"_"
                                   ,repi,".out"), what = "all")
Err1     <- sum(grep("NON-POSITIVE", Out1$errors))
              + sum(grep("DID NOT TERMINATE", Out1$errors))
Err2      <- sum(grep("NON-POSITIVE", Out2$errors))
                 + sum(grep("DID NOT TERMINATE", Out2$errors))
War1     <- length(Out1$warnings)
War2      <- length(Out2$warnings)
X    <- as.matrix(MyData[[itn]][[repi]][,c("X2","X3")])
if (Err1 == 0) {
  if (War1 != 0) {
    if (length(grep("BEST LOGLIKELIHOOD
            VALUE WAS NOT REPLICATED", Out1$warnings))!=0) {
      LCR1mod       <- LCRSurvExtraNoX1(BinData[,names1])
      Fit1          <- mplusModeler(LCR1mod, modelout =
        paste0("LCR1_S",itn,"_",repi,".inp"), run = 1L)
      Out1          <- readModels(paste0("lcr1_s",itn,"_"
      ,repi,".out"), what = "all")
      War1          <- length(Out1$warnings)
      Extra1[repi]  <- 1
```

```
      Fail1[repi]    <- if_else (War1 == 0, 0, 1)
    }
  }
  if (War1 == 0) {
    Beta1  <- as.numeric(Out1$parameters$unstandardized[1:6,"est"])
    Check1 <- as.numeric(Out1$parameters$unstandardized[1:6,"se"])
    Beta1[Check1==0]<- 0
    if (sum(is.na(Beta1))==0 & length(Check1)!=0) {
      LCR1data       <- tibble(Fit1$results$savedata)
      ###################################
      LCR1data$DTH   <- 1 - LCR1data$DTH
      ###################################
      C1             <- C_index(Beta1, LCR1data)
      LCR1[repi]     <- if_else(C1 >= 0.5, C1, 1 - C1)
    } else LCR1[repi] <- NA
  }
} else LCR1[repi]    <- NA
if (Err2 == 0) {
  if (War2 != 0) {
    if (length(grep("BEST LOGLIKELIHOOD
          VALUE WAS NOT REPLICATED", Out2$warnings))!=0) {
      LCR2mod        <- LCRSurvExtraNoX1(BinData[,names1])
      Fit2           <- mplusModeler(LCR2mod,
       modelout = paste0("LCR2_S",itn,"_",repi,".inp"), run = 1L)
      Out2           <- readModels(paste0("lcr2_s",itn,"_",repi,"
              .out"), what = "all")
      War2           <- length(Out2$warnings)
      Extra2[repi]   <- 1
      Fail2[repi]    <- if_else (War2 == 0, 0, 1)
    }
  }
  if (War2 == 0) {
    Beta2  <- as.numeric(Out2$parameters$unstandardized[1:6,"est"])
    Check2 <- as.numeric(Out2$parameters$unstandardized[1:6,"se"])
    Beta2[Check2==0]<- 0
    if (sum(is.na(Beta2))==0 & length(Check2)!=0) {
      LCR2data       <- tibble(Fit2$results$savedata)
      ###################################
      LCR2data$DTH   <- 1 - LCR2data$DTH
      ###################################
      C2             <- C_index(Beta2, LCR2data)
      LCR2[repi]     <- if_else(C2 >= 0.5, C2, 1 - C2)
    } else LCR2[repi] <- NA
```

```
      }
    } else LCR2[repi]    <- NA
  }
  Update    <- data.frame(Scenario=itn+3,Model=1:reps,CoxC1=CoxC1
  ,LCR1=LCR1,Extra1=Extra1,Fail1=Fail1,CoxC2=CoxC2
  ,LCR2=LCR2,Extra2=Extra2,Fail2=Fail2)
  Results                <- bind_rows(Results, Update)
}; print("Total runtime"); print(Sys.time()-t0)
```

References

L. S. Aiken, S. G. West, S. C. Pitts, A. N. Baraldi, and I. C. Wurpts. Multiple linear regression. *Handbook of Psychology, Second Edition*, 2, 2012. 2

P. D. Allison. Change scores as dependent variables in regression analysis. *Sociological methodology*, pages 93–114, 1990. 75

K. F. Arnold, V. Davies, M. de Kamps, P. W. Tennant, J. Mbotwa, and M. S. Gilthorpe. Reflection on modern methods: generalized linear models for prognosis and intervention—theory, practice and implications for machine learning. *International journal of epidemiology*, 49(6):2074–2082, 2020. 5, 99

R. Bender, T. Augustin, and M. Blettner. Generating survival times to simulate cox proportional hazards models. *Statistics in medicine*, 24(11):1713–1723, 2005. 47

J. M. Bland and D. G. Altman. Statistics notes: bootstrap resampling methods. *bmj*, 350, 2015. 107

G. Brassington. Mean absolute error and root mean square error: which is the better metric for assessing model performance? In *EGU General Assembly Conference Abstracts*, page 3574, 2017. 85

G. Celeux and G. Soromenho. An entropy criterion for assessing the number of clusters in a mixture model. *Journal of classification*, 13(2):195–212, 1996. 106

T. Chai and R. R. Draxler. Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, 7(3):1247–1250, 2014. 85

T. Ching, X. Zhu, and L. X. Garmire. Cox-nnet: an artificial neural network method for prognosis prediction of high-throughput omics data. *PLoS computational biology*, 14(4):e1006076, 2018. 37, 140, 143

G. Cochran, V. Hruschak, J. L. Bacci, K. C. Hohmeier, and R. Tarter. Behavioral, mental, and physical health characteristics and opioid medication misuse among community pharmacy patients: A latent class analysis. *Research in Social and Administrative Pharmacy*, 13(6):1055–1061, 2017. 7

D. R. Cox. Partial likelihood. *Biometrika*, 62(2):269–276, 1975. 20

M. J. Crowther and P. C. Lambert. Simulating biologically plausible complex survival data. *Statistics in medicine*, 32(23):4118–4134, 2013. 47

R. M. Cubbon, C. P. Gale, L. C. Kearney, C. B. Schechter, W. P. Brooksby, J. Nolan, K. A. Fox, A. Rajwani, W. Baig, D. Groves, et al. Changing characteristics and mode of death associated with chronic heart failure caused by left ventricular systolic dysfunction: a study across therapeutic eras. *Circulation: Heart Failure*, 4(4):396–403, 2011. 103

G. Currie and C. Delles. Precision medicine and personalized medicine in car-

diovascular disease. *Sex-Specific Analysis of Cardiovascular Function*, pages 589–605, 2018. 4

N. Dean and A. E. Raftery. Latent class analysis variable selection. *Annals of the Institute of Statistical Mathematics*, 62(1):11, 2010. 100

A. P. DeFilippis, R. Young, C. J. Carrubba, J. W. McEvoy, M. J. Budoff, R. S. Blumenthal, R. A. Kronmal, R. L. McClelland, K. Nasir, and M. J. Blaha. An analysis of calibration and discrimination among multiple cardiovascular risk scores in a modern multiethnic cohort. *Annals of internal medicine*, 162(4): 266–275, 2015. 99

F. X. Diebold. On the origin (s) and development of the term'big data'. 2012. 99

A. R. T. Donders, G. J. Van Der Heijden, T. Stijnen, and K. G. Moons. A gentle introduction to imputation of missing values. *Journal of clinical epidemiology*, 59(10):1087–1091, 2006. 5

S. Dreiseitl and L. Ohno-Machado. Logistic regression and artificial neural network classification models: a methodology review. *Journal of biomedical informatics*, 35(5-6):352–359, 2002. 3

G. T. Ellison. Might temporal logic improve the specification of directed acyclic graphs (dags)? *Journal of Statistics and Data Science Education*, (just-accepted):1–18, 2021. 67

D. Faraggi and R. Simon. A neural network model for survival data. *Statistics in medicine*, 14(1):73–82, 1995. 142

S. C. Gadd, P. W. Tennant, A. J. Heppenstall, J. R. Boehnke, and M. S. Gilthorpe. Analysing trajectories of a longitudinal exposure: A causal perspective on common methods in lifecourse research. *PloS one*, 14(12):e0225217, 2019. 101

J. Gaudart, B. Giusiano, and L. Huiart. Comparison of the performance of multilayer perceptron and linear regression for epidemiological data. *Computational statistics & data analysis*, 44(4):547–570, 2004. 2

D. Gefen, D. Straub, and M.-C. Boudreau. Structural equation modeling and regression: Guidelines for research practice. *Communications of the association for information systems*, 4(1):7, 2000. 25

M. Gilthorpe, D. Dahly, Y.-K. Tu, L. Kubzansky, and E. Goodman. Challenges in modelling the random structure correctly in growth mixture models and the impact this has on model mixtures. *Journal of developmental origins of health and disease*, 5(3):197–205, 2014. 106

M. S. Gilthorpe, W. J. Harrison, A. Downing, D. Forman, and R. M. West. Multilevel latent class casemix modelling: a novel approach to accommodate patient casemix. *BMC health services research*, 11(1):1–7, 2011. 20, 21

I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016. 32

S. W. Grant, G. S. Collins, and S. A. Nashef. Statistical primer: developing and validating a risk prediction model. *European Journal of Cardio-Thoracic Surgery*, 54(2):203–208, 2018. 1

K. J. Grimm, G. L. Mazza, and P. Davoudzadeh. Model selection in finite mixture

models: A k-fold cross-validation approach. *Structural Equation Modeling: A Multidisciplinary Journal*, 24(2):246–256, 2017. 106

K. Hajian-Tilaki. Receiver operating characteristic (roc) curve analysis for medical diagnostic test evaluation. *Caspian journal of internal medicine*, 4(2):627, 2013. 107

F. E. Harrell, R. M. Califf, D. B. Pryor, K. L. Lee, and R. A. Rosati. Evaluating the yield of medical tests. *Jama*, 247(18):2543–2546, 1982. 42, 145

F. E. Harrell Jr, K. L. Lee, R. M. Califf, D. B. Pryor, and R. A. Rosati. Regression modelling strategies for improved prognostic prediction. *Statistics in medicine*, 3(2):143–152, 1984. 42, 145

F. E. Harrell Jr, K. L. Lee, and D. B. Mark. Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in medicine*, 15(4):361–387, 1996. 43

W. J. Harrison, M. S. Gilthorpe, A. Downing, and P. D. Baxter. Multilevel latent class modelling of colorectal cancer survival status at three years and socioeconomic background whilst incorporating stage of disease. *International Journal of Statistics and Probability*, 2(3):85, 2013. 21, 106, 107

P. J. Heagerty, T. Lumley, and M. S. Pepe. Time-dependent roc curves for censored survival data and a diagnostic marker. *Biometrics*, 56(2):337–344, 2000. 107

J. M. Hendriksen, G.-J. Geersing, K. G. Moons, and J. A. de Groot. Diagnostic

and prognostic prediction models. *Journal of Thrombosis and Haemostasis*, 11: 129–141, 2013. 3

K. R. Hess. Graphical methods for assessing violations of the proportional hazards assumption in cox regression. *Statistics in medicine*, 14(15):1707–1723, 1995. 19

C. Hitchcock and E. Sober. Prediction versus accommodation and the risk of overfitting. *British Journal for the Philosophy of Science*, 55(1), 2004. 105

R. R. Hocking. A biometrics invited paper. the analysis and selection of variables in linear regression. *Biometrics*, pages 1–49, 1976. 5

C. Holmberg and M. Parascandola. Individualised risk estimation and the nature of prevention. *Health, risk & society*, 12(5):441–452, 2010. 99

A. Huitfeldt. Is caviar a risk factor for being a millionaire? *bmj*, 355, 2016. 101

J. Jerez, L. Franco, E. Alba, A. Llombart-Cussac, A. Lluch, N. Ribelles, B. Munarriz, and M. Martin. Improvement of breast cancer relapse prediction in high risk intervals using artificial neural networks. *Breast Cancer Research and Treatment*, 94(3):265–272, 2005. 141, 142

B. Karlik and A. V. Olgac. Performance analysis of various activation functions in generalized mlp architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems*, 1(4):111–122, 2011. 36

O. Khan, J. H. Badhiwala, G. Grasso, and M. G. Fehlings. Use of machine learning and artificial intelligence to drive personalized medicine approaches for spine care. *World neurosurgery*, 140:512–518, 2020. 140

A. M. Killu, C. B. Granger, and B. J. Gersh. Risk stratification for stroke in atrial fibrillation: a critique. *European heart journal*, 40(16):1294–1302, 2019. 99

D. M. Kline and V. L. Berardi. Revisiting squared-error and cross-entropy functions for training neural network classifiers. *Neural Computing & Applications*, 14(4):310–318, 2005. 34

L. D. Kubzansky, M. S. Gilthorpe, and E. Goodman. Erratum to: A prospective study of psychological distress and weight status in adolescents/young adults. *Annals of Behavioral Medicine*, 48(2):284–285, 2014. 21

S. Kuhle, B. Maguire, H. Zhang, D. Hamilton, A. C. Allen, K. Joseph, and V. M. Allen. Comparison of logistic regression with machine learning methods for the prediction of fetal growth abnormalities: a retrospective cohort study. *BMC pregnancy and childbirth*, 18(1):1–9, 2018. 140

M. Kuhn, K. Johnson, et al. *Applied predictive modeling*, volume 26. Springer, 2013. 107

A. Y. Kuk. All subsets regression in a proportional hazards model. *Biometrika*, 71(3):587–592, 1984. 106

E. R. Lee, H. Noh, and B. U. Park. Model selection via bayesian information criterion for quantile regression models. *Journal of the American Statistical Association*, 109(505):216–229, 2014. 81

E. T. Lee and O. T. Go. Survival analysis in public health research. *Annual review of public health*, 18(1):105–134, 1997. 23, 66

J. Li, J.-h. Cheng, J.-y. Shi, and F. Huang. Brief introduction of back propagation (bp) neural network algorithm and its improvement. In *Advances in computer science and information engineering*, pages 553–558. Springer, 2012. 34

W. Li and D. R. Nyholt. Marker selection by akaike information criterion and bayesian information criterion. *Genetic Epidemiology*, 21(S1):S272–S277, 2001. 81

J. Magidson and J. K. Vermunt. Latent class analysis. *The Sage handbook of quantitative methodology for the social sciences*, pages 175–198, 2004. 7

J. N. Mandrekar. Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*, 5(9):1315–1316, 2010. 116, 119

J. L. Mbotwa, M. d. Kamps, P. D. Baxter, G. T. Ellison, and M. S. Gilthorpe. Latent class regression improves the predictive acuity and clinical utility of survival prognostication amongst chronic heart failure patients. *Plos one*, 16 (5):e0243674, 2021. 123, 124

C. E. Metz. Basic principles of roc analysis. In *Seminars in nuclear medicine*, volume 8, pages 283–298. Elsevier, 1978. 107

T. P. Morris, I. R. White, and M. J. Crowther. Using simulation studies to evaluate statistical methods. *Statistics in medicine*, 38(11):2074–2102, 2019. 76

L. K. Muthén and B. O. Muthén. 1998–2012. mplus user's guide. *Los Angeles: Muthén & Muthén*, 2012. 21

K. F. Nimon and F. L. Oswald. Understanding the results of multiple linear regression: Beyond standardized regression coefficients. *Organizational Research Methods*, 16(4):650–674, 2013. 81

K. L. Nylund, T. Asparouhov, and B. O. Muthén. Deciding on the number of classes in latent class analysis and growth mixture modeling: A monte carlo simulation study. *Structural equation modeling: A multidisciplinary Journal*, 14(4):535–569, 2007. 21, 105

L. Ohno-Machado. A comparison of cox proportional hazards and artificial neural network models for medical prognosis. *Computers in biology and medicine*, 27 (1):55–65, 1997. 143

L. Ohno-Machado. Modeling medical prognosis: survival analysis techniques. *Journal of biomedical informatics*, 34(6):428–439, 2001. 3, 141

C. J. O'Donnell. Opportunities and challenges for polygenic risk scores in prognostication and prevention of cardiovascular disease. *JAMA cardiology*, 5(4): 399–400, 2020. 99

N. Papachristou, C. Miaskowski, P. Barnaghi, R. Maguire, N. Farajidavar, B. Cooper, and X. Hu. Comparing machine learning clustering with latent class analysis on cancer symptoms' data. In *2016 IEEE Healthcare Innovation Point-Of-Care Technologies Conference (HI-POCT)*, pages 162–166. IEEE, 2016. 7

J. Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995. 29

J. Pearl and T. S. Verma. A theory of inferred causation. In *Studies in Logic and*

the Foundations of Mathematics, volume 134, pages 789–811. Elsevier, 1995. 67

M. Piccininni, S. Konigorski, J. L. Rohmann, and T. Kurth. Directed acyclic graphs and causal thinking in clinical risk prediction modeling. *BMC medical research methodology*, 20(1):1–9, 2020. 8, 73

S. Prinja, N. Gupta, and R. Verma. Censoring in clinical trials: review of survival analysis techniques. *Indian journal of community medicine: official publication of Indian Association of Preventive & Social Medicine*, 35(2):217, 2010. 17

J. G. Richens, C. M. Lee, and S. Johri. Improving the accuracy of medical diagnosis with causal machine learning. *Nature communications*, 11(1):1–9, 2020. 8, 73

B. Rockhill, I. Kawachi, and G. A. Colditz. Individual risk prediction and population-wide disease prevention. *Epidemiologic Reviews*, 2000. 99, 101

S. Sabouri, H. Esmaily, S. Shahidsales, and M. Emadi. Survival prediction in patients with colorectal cancer using artificial neural network and cox regression. *International Journal of Cancer Management*, 13(1), 2020. 2

S. Senn. Change from baseline and analysis of covariance revisited. *Statistics in medicine*, 25(24):4334–4344, 2006. 54, 75

E. Shahar and D. J. Shahar. Causal diagrams and change variables. *Journal of evaluation in clinical practice*, 18(1):143–148, 2012. 53

N. Shahid, T. Rappon, and W. Berta. Applications of artificial neural networks

in health care organizational decision-making: A scoping review. *PloS one*, 14 (2):e0212356, 2019. 32

S. Sharma. Activation functions in neural networks. *Towards Data Science*, 6, 2017. 35

A. Skrondal and S. Rabe-Hesketh. *Generalized latent variable modeling: Multilevel, longitudinal, and structural equation models*. Crc Press, 2004. 3

X. Song, A. Mitnitski, J. Cox, and K. Rockwood. Comparison of machine learning techniques with classical statistical models in predicting health outcomes. In *Medinfo*, pages 736–740, 2004. 7

M. Sperrin and B. McMillan. Prediction models for covid-19 outcomes, 2020. 6

M. Sperrin, G. P. Martin, A. Pate, T. Van Staa, N. Peek, and I. Buchan. Using marginal structural models to adjust for treatment drop-in when developing clinical prediction models. *Statistics in medicine*, 37(28):4142–4154, 2018. 6

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 42

E. W. Steyerberg and Y. Vergouwe. Towards better clinical prediction models: seven steps for development and an abcd for validation. *European heart journal*, 35(29):1925–1931, 2014. 2

P. Tarka. An overview of structural equation modeling: its beginnings, historical development, usefulness and controversies in the social sciences. *Quality & quantity*, 52(1):313–354, 2018. 26

P. Tennant, K. Arnold, L. Berrie, G. Ellison, and M. Gilthorpe. Advanced modelling strategies: challenges and pitfalls in robust causal inference with observational data. In *Advanced Modelling Strategies: Challenges and pitfalls in robust causal inference with observational data*. Leeds Institute for Data Analytics, 2017. 5

P. W. Tennant, K. F. Arnold, G. T. Ellison, and M. S. Gilthorpe. Analyses of'change scores' do not estimate causal effects in observational data. *International journal of epidemiology*, pages 1–12, 2021a. 13, 53, 54, 75

P. W. Tennant, E. J. Murray, K. F. Arnold, L. Berrie, M. P. Fox, S. C. Gadd, W. J. Harrison, C. Keeble, L. R. Ranker, J. Textor, et al. Use of directed acyclic graphs (dags) to identify confounders in applied health research: review and recommendations. *International journal of epidemiology*, 50(2):620–632, 2021b. 100

J. Textor, B. van der Zander, M. S. Gilthorpe, M. Liśkiewicz, and G. T. Ellison. Robust causal inference using directed acyclic graphs: the r package 'dagitty'. *International journal of epidemiology*, 45(6):1887–1894, 2016. 68

J. B. Ullman and P. M. Bentler. Structural equation modeling. *Handbook of Psychology, Second Edition*, 2, 2012. 25

M.-C. Wang, Q. Deng, X. Bi, H. Ye, and W. Yang. Performance of the entropy as an index of classification accuracy in latent profile analysis: a monte carlo simulation study. *Acta Psychologica Sinica*, 2017. 106

B. E. Weller, N. K. Bowen, and S. J. Faubert. Latent class analysis: a guide to best practice. *Journal of Black Psychology*, 46(4):287–311, 2020. 105

C. E. Werts and R. L. Linn. A general linear model for studying growth. *Psychological Bulletin*, 73(1):17, 1970. 75

D. Westreich and S. Greenland. The table 2 fallacy: presenting and interpreting confounder and modifier coefficients. *American journal of epidemiology*, 177 (4):292–298, 2013. 118, 120

J. Wilkinson, K. F. Arnold, E. J. Murray, M. van Smeden, K. Carr, R. Sippy, M. de Kamps, A. Beam, S. Konigorski, C. Lippert, et al. Time to reality check the promises of machine learning-powered precision medicine. *The Lancet Digital Health*, 2020. 4, 99

K. K. Witte, M. Drozd, A. M. Walker, P. A. Patel, J. C. Kearney, S. Chapman, R. J. Sapsford, J. Gierula, M. F. Paton, J. Lowry, et al. Mortality reduction associated with β-adrenoceptor inhibition in chronic heart failure is greater in patients with diabetes. *Diabetes Care*, 41(1):136–142, 2018a. 103

K. K. Witte, P. A. Patel, A. M. Walker, C. B. Schechter, M. Drozd, A. Sengupta, R. Byrom, L. C. Kearney, R. J. Sapsford, M. T. Kearney, et al. Socioeconomic deprivation and mode-specific outcomes in patients with chronic heart failure. *Heart*, 104(12):993–998, 2018b. 103

S. Wright. On the nature of size factors. *Genetics*, 3(4):367, 1918. 26

S. Wright. Correlation and causation. 1921a. xv, 26, 28

S. Wright. The method of path coefficients. *The annals of mathematical statistics*, 5(3):161–215, 1934. 26

R. Zemek, N. Barrowman, S. B. Freedman, J. Gravel, I. Gagnon, C. McGahern, M. Aglipay, G. Sangha, K. Boutis, D. Beer, et al. Clinical risk score for persistent postconcussion symptoms among children with acute concussion in the ed. *Jama*, 315(10):1014–1025, 2016. 2

B. Zupan, J. DemšAr, M. W. Kattan, J. R. Beck, and I. Bratko. Machine learning for survival analysis: a case study on recurrence of prostate cancer. *Artificial intelligence in medicine*, 20(1):59–75, 2000. 7