Neural Network Modeling of Time-Dependent Event Probabilities for Health Outcome Prediction

---

Fabio Luis de Mello

25th April 2023
Version: Final Version

Neural Network Modeling of Time-Dependent Event Probabilities for Health Outcome Prediction

BY

Fabio Luis de Mello

A dissertation submitted to

The University of Sheffield

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

25th April 2023

Fabio Luis de Mello

Neural Network Modeling of Time-Dependent Event Probabilities for Health Outcome Prediction

PhD Dissertation, 25th April 2023

Supervisors: Prof Visakan Kadirkamanathan
Prof J Mark Wilkinson

The University of Sheffield

Department of Automatic Control and Systems Engineering

Amy Johnson Building

Portobello Street

Sheffield, S1 3JD

Dedication

To my loving wife who is always there making my life happier and more meaningful.

A solemn consideration, when I enter a great city by night, that every one of those darkly clustered houses encloses its own secret; that every room in every one of them encloses its own secret; that every beating heart in the hundreds of thousands of breasts there, is, in some of its imaginings, a secret to the heart nearest it!

Charles Dickens

Acknowledgement

I thank my supervisors Visakan Kadirkamanathan and Mark Wilkinson, who had numerous invaluable conversations and advice that enabled me to perform this work.

I thank my wife, Camila Pedrosa de Mello, and all my family who always supported me during this great challenge.

I thank the National Joint Registry for my studentship. i thank the patients and staff of all the hospitals in England, Wales and Northern Ireland who have contributed data to the National Joint Registry (NJR). i AM grateful to the Healthcare Quality Improvement Partnership (HQIP), the NJR Research Committee and staff at the NJR Centre for facilitating this work. The author has conformed to the NJR's standard protocol for data access and publication. The views expressed represent those of the author and do not necessarily reflect those of the National Joint Registry Steering Committee or the Healthcare Quality Improvement Partnership (HQIP) who do not vouch for how the information is presented.

Declaration

I, Fabio Luis de Mello, declare that the work presented in this thesis is my own. All material in this thesis which is not of my own work, has been properly accredited and referenced.

Sheffield, 25th April 2023

Fabio Luis de Mello

Abstract

The estimation of surgery outcomes is of great value when deciding if surgery is the best treatment for a particular patient. In the present work, machine learning methods were developed to enable estimation of revision and mortality risks after joint replacements and also the prediction of postoperative PROMs. The estimation of the revision and mortality risks are within the domain of survival analysis which has recently received increased attention in the machine learning literature with the ultimate goal of achieving universal approximation similarly to other machine learning tasks. With that purpose, we proposed the metaparametric neural network (MNN) framework which is a hierarchical alternative to neural networks that allows analytical manipulation of the model function representation. The Universal Approximation Theorem for Metaparametric Neural Networks provides a formal guarantee that the proposed class of models can approximate any continuous function arbitrarily well similarly to other neural network structures. The MNN framework was applied to survival analysis allowing the development of three different models, each being derived from a different class of survival models, proportional hazards (PH-MNN), direct hazards (DH-MNN) and quantile regression (QR-MNN). In particular, the PH-MNN model achieved the best results and outperformed current state-of-the-art. An alternative version of the PH-MNN model, nested PH-MNN, was also proposed achieving similar performance and allowing better interpretability as well as a transfer learning strategy that successfully reduced overfit. In the prediction of postoperative PROMs, the main contribution was casting of the estimation problem as a classification problem rather than a regression problem. This allowed a probabilistic modelling that comprehensively represents the data being modelled, allowing retrieval of other metrics with better interpretability. The proposed approach achieved better estimation performance than state-of-the-art.

Contents

List of Figures

List of Tables

1

Introduction

1.1 Background and context

A joint replacement surgery is a common medical procedure to treat Osteoarthritis and other pathologies. While they rarely impose a direct life threat to patients, these pathologies might have serious consequences in their ability to perform important daily life activities. Similarly to most medical treatments, the outcome of joint replacement surgeries is not certain and in rare cases, they might have severe adverse consequences. As a result, there are several cases in which it is not possible to determine objectively if a surgery is the best treatment and this decision should be largely determined by the patient's perception of the trade off between the potential benefits to be achieved with it and the risks to be taken. In order for the medical decisions to best reflect the interest of the patients, it is important that patients are provided the most accurate possible knowledge about the impacts of their decision. Based on this need, the National Joint Registry (NJR) for England, Wales, Northern Ireland and the Isle of Man has financed the development of a patient decision aid (PDA) web based tool, which allows patients to enter their information and receive a personalized assessment of their prognostics after a hip or a knee replacement. The present PhD project is a part of this development and its main goal was the development of machine learning methods to estimate the outcomes of joint replacements, allowing an improvement of the PDA tool.

There are multiple ways in which the outcome a joint replacement might be different from what is expected. First, it is possible that the desired improvement in the performance of daily activities is not achieved. Also, there are cases in which the joint health worsens again some time after the surgery, being necessary for the patient to undergo a revision surgery. Indeed, according to the dataset from the NJR, which contains data from over 1,00,000 knee and 800,000 hip replacement surgeries performed in these countries from 2003 to 2018, 4.3% and 2.9% of patients who undergo respectively a knee or a hip replacement surgery have a revision within ten years from the first procedure. Even though they are low risk surgeries, there is also a small chance that a patient might die during the surgery or because of a later complication. The risk of death within one month from a joint replacement surgery according to the same dataset is 0.14% and 0.15% respectively after a knee or a hip replacement surgery.

There are several factors that influence the outcome of a surgery, some of which can be easily measured and accounted for like age, BMI and the type of implant used, and some of which might be impossible to completely account for like the ability of the surgeon or how much does the patient comply with medical recommendations. Although it might be impossible to identify and account for all factors that may influence the outcome of a surgery, it is important to do so to the maximum possible extent. Indeed, complications that are rare for the average population might be more common within specific groups of patients and identifying this type of pattern is fundamental for improving the decision process about whether or not a surgery is the best treatment for a given patient.

The PDA tool which is being financed by the NJR intends to overcome these difficulties in the collaboration between doctor and patient to achieve a decision about the treatment that best reflects the interest of the patient. In this tool, a web interface is made available to patients, where they can fill their information either by themselves or with the help of a doctor. In the later case, a more detailed model is available. As a result, patients are informed of their expected post operative joint health, which is measured by the patient reported outcomes measures (PROMs). Additionally, they are informed of the probability for having to redo the surgery in the following years and the risk of death throughout the year following the surgery. Before this project the PDA tool used linear models to estimate each outcome of the surgery, having margin for improvement with the usage of machine learning methods. The present PhD project is

part of this development and has investigated novel machine learning methods to improve the performance of the PDA tool.

1.2 Research aims and objectives

The estimation of the risks of revision and death after a surgery are within the theoretical framework of survival analysis, which studies methods that estimate, as a function of the observation time, the probability for some event to have already occurred. Survival analysis can be interpreted as an extension of both regression and classification modeling scenarios. In regression problems, it is necessary to estimate a real valued quantity as a function of input variables. Survival analysis extends this domain by modeling instead the probability distribution for the same quantity. In classification problems, it is necessary to estimate how the probability of a label depends on the input variables. This domain is extended in survival analysis by making the label probability time dependent. While the domain of survival analysis models is more general than the domain of regression and classification models, this imposes additional challenges to it, which are usually solved by imposing simplifying hypothesis that restrict the models to a particular class of problems.

The most traditional approach for developing survival models, in which is based the model currently used in the PDA tool, is the proportional hazards assumption. It divides the survival model in two parts, one exclusively time dependent and other exclusively dependent on the input variables, usually known as covariates. Within this assumption, the time dependent component of the probability distribution is the same for all possible values of the covariates and the component that is dependent on the covariates has a one dimensional output, allowing the estimation of the model parameters with the available data. While this assumption makes the estimation process easier, there is no guarantee that the true survival probability of an experimental scenario obeys it. Indeed, there are numerous practical examples in the literature of circumstances in which it is not possible to achieve a satisfactory model within the proportional hazards assumption.

The recent advances in machine learning literature has led to unprecedented results in statistical modeling task in the presence of big data. The purpose of the current project is to explore to the maximum possible extent the machine

learning techniques in the tasks present in the PDA tool. While more generic models usually require more data to be estimated than less generic models, this allows the resulting model to achieve a more precise description of the studied phenomenon. Indeed, a more generic model is capable of producing a broader class of functions, which means that more data is necessary to choose between all possible functions that the model can represent and also that this class of functions is more likely to contain a good approximation of the true function that is being estimated. The improvement of the estimation accuracy of the models in the PDA tool leads to a more accurate knowledge by the patients about the consequences of a joint replacement surgery, which provides them a better basis to make decisions about the type of treatment they will undergo. Nevertheless, while the estimations of post-operative PROMs is within the scope of well established machine learning methods, there are important challenges to be overcome in the application of machine learning to survival analysis. The present work has improved the state of the art for machine learning in survival analysis, and a class of models with universal approximation property was proposed.

The most important aspects to be considered in the development of a survival analysis model are the parametrization of the model and the objective function used for parameter estimation. For the first problem, the three most relevant approaches in the statistical literature are the proportional hazards model, the accelerated failure time (AFT) model and the discrete time model. The AFT model supposes that the failure process is accelerated or decelerated depending on the covariates without changes in the shape of the failure probability distribution. The discrete time models split the follow up time into small intervals, allowing the approximation of the estimates in each interval to standard classification methods. Another aspect that is important to the development of survival models is the extension to competing risks scenarios. This is a type of scenario in which there are more than one event of interest, which are mutually exclusive. Although prior to this work, the models in the PDA tool were all single risk, the possibility to use competing risks models to estimate the probability of different causes of death or different causes of revision will be studied in this work.

The tasks to be performed in the present work in order to allow the improvement of the PDA tool are:

a) Development of a generic machine learning method for survival mod-

eling.

b) Establishment of an estimation procedure for the proposed model, allowing parameters to be successfully estimated from data.

c) Competing risks extension of the proposed method.

d) Application of the proposed method to estimation of revision and mortality risks after hip and knee replacement surgeries.

e) Development of a machine learning model for estimation of postoperative PROMs.

1.3 Research Contributions

The main contributions of this work are the following:

a) Proposal of the metaparametric neural network (MNN) framework for machine learning allowing the extension of a broad range of survival models and providing better interpretability of the estimated model due to its grey-box structure in opposition to the black-box structure of vanilla neural networks.

b) Proof of the Universal Approximation Theorem for MNNs, which guarantees that MNNs can approximate any continuous function provided that enough parameters are provided.

c) Proof of Theorem 2, which enables estimation of the PH-MNN model with the profile likelihood and shows that the couple baseline hazard model class (to which the PH-MNN model belongs) can achieve the maximum likelihood among all possible survival models.

d) Proof of Theorem 3 providing the theoretical background for the partial likelihood estimation of the PH-MNN model.

e) Proposal of a transfer learning strategy allowing the use of heterogeneous data to avoid overfit.

f) Novel application of classifier neural networks to perform probabilistic estimation of postoperative PROMs scores.

1.4 Thesis Outline

The chapters of this thesis are organized as follows:

- Chapter 1 - Introduction: the present chapter, which included the background for the work, the research aims and objectives, the list of contributions and the outline of the thesis.
- Chapter 2 - Literature Review: a detailed description of the state of the art of survival modeling is provided including both statistical and machine learning methods.
- Chapter 3 - Metaparametric Neural Networks: a novel class of machine learning models is proposed and a universal approximation theorem is proven showing that these models can approximate any continuous functions over a compact space.
- Chapter 4 - Metaparametric Neural Networks for Survival Analysis: The MNN modeling approach is applied to survival analysis, resulting in the extension of multiple classes of survival models to achieve the universal approximation property.
- Chapter 5 - Maximum Likelihood Derivation of the PH-MNN Estimator and its Properties: The proof of two theorems is presented. The first showing that the coupled baseline hazard structure (which includes the PH-MNN model) can achieve the maximum likelihood among all possible survival models. The second showing that partial likelihood estimation achieves the same asymptotic properties as profile likelihood estimation.
- Chapter 6 - Survival Modeling of Joint Replacement Surgeries: The nested PH-MNN model is applied to estimate revision and mortality risks after hip and knee replacements. A transfer learning strategy is proposed to allow the use of data without preoperative PROMs input as part of the training of a model that includes preoperative PROMs, reducing overfit.
- Chapter 7 - Neural Network Classifier Approach for Postoperative PROMs Prediction: The task of estimating postoperative PROMs was cast as a classification task instead of regression and a classifier neural

network was used to predict postoperative PROMs, achieving better performance than current state of the art.

- Chapter 8 - Conclusions and Future Work: The results of the work are summarized and analyzed with reference to the research aims and objectives. Possible directions for future research are pointed out.

2

Literature Review

Survival analysis studies mathematical models of scenarios where subjects are expected to experience a given event at a time which is unknown prior to observation. The most common application of survival analysis is in medical studies, where it is desired to determine the risk of death or the risk of a certain disease as a function of time within a group of patients.

2.1 Basic concepts

2.1.1 Mathematical representations of the survival probability

The main task in survival analysis is to model the occurrence of events as function of time. In the simplest case in which only one type of event is studied, there is only one random variable being modeled which is the time for the first occurrence of the event. Then, the estimation problem can be seen as the estimation of the probability distribution of this random variable $f(t)$. There are two main differences between this and other scenarios where the probability distribution of a real-valued outcome is estimated. First, that the time horizon for observation of the event is intrinsically limited because no experimental setting would allow infinite observation. Second, that there are scenarios in which the event of interest will never be observed. Therefore, in that case the event time

probability distribution does not rigorously follow the definition of a probability distribution since its integral is smaller than 1. In practice, this possibility is typically ignored in the literature since the time horizon of models is limited and the event time could simply be larger than the observation horizon. Nonetheless, there are cases in which an event would never be observed for some subject even if observed for an indefinite amount of time. In these cases, the mathematical rigor can be reestablished by defining a Boolean random variable E which is 1 if the event happens to the subject of interest in the studied time horizon and 0 otherwise. Then, it is possible to study the joint probability of E and t in the form $f(t, E = 1) = f(t|E)P(E)$. Note that this joint probability distribution is equal to $f(t)$ within the studied time horizon as long as it's assumed that if an event is not observed in the time horizon it will be observed later.

Because of the intrinsically limited observation time horizon, the time to event is typically not modeled directly in the form of a probability distribution, but in an alternative representation. The following representations of the time to event probability distribution are defined in [1], Section 1.4, p. 13:

- **Survival function**: $S(t)$ is the probability for the studied event not happening to a given subject until time t. It is associated to $f(t)$ through the expression:

$$S(t) = 1 - \int_0^t f(t)(d)t\,. \tag{2.1}$$

- **Incidence function**: $F(t)$ is the probability for the event to happen to a given subject until time t. It is associated to $S(t)$ through the expression:

$$F(t) = 1 - S(t)\,. \tag{2.2}$$

- **Hazard function**: $\lambda(t)$ is the event probability density at time t given that it has not happened yet to the subject of interest. It was defined in [2] and is essential to the definition of the proportional hazards model that will be detailed in Section 2.2.2. It is associated to $S(t)$ through the expression:

$$\lambda(t) = -\frac{1}{S(t)}\frac{\mathrm{d}}{\mathrm{d}t}S(t) = -\frac{\mathrm{d}}{\mathrm{d}t}\ln S(t)\,. \tag{2.3}$$

- **Cumulative hazard function**: $\Lambda(t)$ is the time integration of $\lambda(t)$:

$$\Lambda(t) = \int_0^t \lambda(\nu)\mathrm{d}\nu\,. \tag{2.4}$$

  It is associated to $S(t)$ through the expression:

$$S(t) = \exp[-\Lambda(t)]\,. \tag{2.5}$$

The survival probability may be modeled either as being the same for the entire population or as depending on a set of features from the subject, which are called covariates and denoted by $\mathbf{x} \in \mathbb{R}^{N_x}$. If the survival probability is modeled as a function of the covariates, its mathematical representations become: $S(t;\mathbf{x})$, $F(t;\mathbf{x})$, $\lambda(t;\mathbf{x})$ and $\Lambda(t;\mathbf{x})$. There, the covariates $\mathbf{x}$ are not modeled as random variables but treated as parameters of the time to event probability distribution.

In some applications, it is necessary to model multiple events. These are called competing risks scenarios. There, the random variables involved would be the time for each of the events to happen and the target of the survival model is to estimate their joint distribution. Similarly to the single risk case, this scenario is not modeled through a probability distribution but through one of the following equivalent representations [3]:

- **Cause specific hazard function**: defined by [3] as the instantaneous probability density for event j at time t given that it has not happened yet for the subject of interest:

$$\lambda_j(t,\mathbf{x}) = \lim_{\Delta t \to 0^+} Pr\{t \leq T < t + \Delta t, J = j | T \geq t; \mathbf{x}\}/\Delta t\,, \tag{2.6}$$

  where Pr denotes a probability, T is the time when the first event was observed for the subject of interest and J is the event type that was observed at time T. It is possible to retrieve the overall survival function from $\lambda_j(t,\mathbf{x})$ through the expression:

$$S(t,\mathbf{x}) = \exp\left(-\int_0^t \lambda(\nu,\mathbf{x})\mathrm{d}\nu\right) = \exp\left(-\int_0^t \sum_j \lambda_j(\nu,\mathbf{x})\mathrm{d}\nu\right)\,. \tag{2.7}$$

- **Cumulative Incidence Function**: defined by [4] as the probability for an

specific event to happen until time t. It is given by:

$$F_j(t, \mathbf{x}) = Pr\{T \leq t, J = j|\mathbf{x}\}\,. \tag{2.8}$$

It can be retrieved from the cause specific hazard function:

$$F_j(t, \mathbf{x}) = \int_0^t S(\nu, \mathbf{x})\lambda_j(\nu, \mathbf{x})\mathrm{d}\nu\,. \tag{2.9}$$

2.1.2 Data censoring

A subject is considered to be censored when the event of interest is not observed during the follow up time and the subject ceases to be observed after it. This means that when censoring is present, the event time data is not completely available. The relationship between censoring and the events of interest are not uniform throughout all applications since it depends on the underlying phenomenon that causes censoring. For example, if censoring happens because the experimental setting imposes a limit to the maximum follow up time of a subject, then the censoring has not relationship to the event being modeled. Conversely, if censoring is caused by an event that happens to a subject, for example if a subject withdraws from a medical study, then it might be statistically related to the event of interest or not. However, in survival analysis literature, the censoring events are most often considered to be statistically independent from the event of interest.

If censoring is not properly taken into account in estimating a survival model, it can lead to biased models. As a result, it is important that survival models are capable of dealing properly with censoring. Two different classifications of censoring are given in [5]. First, the censoring can be classified according to the degree of stochasticity (i.e. whether the censoring time can be inferred deterministically from some hard criteria used in data collection or if it depends on factors that are not controlled when data is being collected requiring it to be treated as a random phenomenon):

- **Type I censoring**: the maximum amount of time that a subject will be observed is fixed prior to the experiment. This happens for example when the data collection period in a study has a fixed length.
- **Type II censoring**: the observation stops after a predefined number of

events have been observed. Here, the censoring time is a random variable that is correlated to the survival time.

- **Random censoring**: the observation suffers from interruptions that cannot be controlled because they depend on: the capability of the research team to keep track of all subjects in the study; or on the voluntary decision of patients in a medical study; or on greater force reasons that stops the observation. Here, the censoring time is a random variable that might be correlated or not to the survival time. In most applications it is assumed that censoring and survival times are statistically independent.

Second, the censoring can be classified according to the temporal relationship to the event of interest:

- **Right censoring**: a subject stops being observed before the studied event is observed. Making the assumption that the subject will never experience the event would be incorrect, since this is not what the data is informing. Indeed, this assumption would lead to a bias for estimating the probability distribution to be lower than it actually is.

- **Left censoring**: the subject starts being observed after it has already been under risk of experiencing the event for some time and some subjects might already have observed the event when the observation begins. While in this case, the event time for some subjects is known, it is possible that there were other subjects that were not observed for some period and have experienced the event before they started being observed. If left censored subjects were treated indistinguishably from others, this would lead to lead to a bias for estimating the event probability to be lower than it actually is.

- **Interval censoring**: the event was experienced by a subject a within known time period, but the event time is not known.

In practice, most survival models only deal with random or type I right censoring, since it is inevitable in most applications. Indeed, by the time a study is finished, it is rarely the case that all the subjects being studied have already experienced the event. On the other hand, left, interval or type II censoring can be avoided in most applications at the time when the data is being collected.

2.1.3 Modelling approaches in survival analysis

A generic survival model shall be able to assign an arbitrary survival function to any set of input covariates. It is common for survival functions to not belong to a restricted family of probability distributions like exponential, Weibull or gamma. Therefore, a generic survival model shall be able to assign a survival function with any arbitrary shape to any set of input covariates; and, in principle, this shape may depend on the value of the covariates. At the same time, the data used to train a survival model consists of realizations of the survival distribution, which are either event times or censoring times. As a result, a single observation provides little information about the shape of the survival function. This usually requires that some simplifying assumption is made to the survival model, so that patterns in the survival function that are common to the entire population do not have to be estimated repeatedly for different values of the covariates. One of the greatest challenges in developing a survival model is to adequately choose those assumptions so that the model is flexible enough to model any experimental scenario. In recent years, the usage of machine learning methods for survival analysis has increased. This is done primarily with the intention of developing more general survival models that are capable of representing patterns that cannot be represented by statistical survival models. There are three different approaches in the literature for applying machine learning to survival analysis. First, by extending the statistical frameworks, so that the models can represent more general patterns while still using the statistical simplifying hypothesis. Second, by dividing the problems into several simpler problems in which only part of the dataset can be used to train each simpler model. Third, by using a generative approach that allows sampling from the survival probability distribution, but does not provide an explicit representation of it.

2.2 Statistical frameworks for survival analysis

The statistical frameworks for survival analysis are built upon assumptions about the class of possible survival models. These frameworks are accompanied by strong mathematical foundation for the estimation procedure. Nevertheless, the assumptions upon which they are built restrict the type of patterns that they are capable of representing. Several approaches to make these frameworks more generic can be found in the literature. Machine learning models built within these frameworks focus specially in allowing the representation of non-

linear patterns. Despite all the effort, none of the extensions proposed so far allow the representation of a completely generic survival analysis scenario. In this section, the most important statistical frameworks for survival analysis and their machine learning extensions are described.

2.2.1 Entire population model

In some applications, the survival probability is considered to be the same for an entire population. This is done either because the knowledge of how the survival function varies among individuals is unnecessary or because the amount of data available is not enough to produce a model that depends on the input covariates. Additionally, there are situations in which an entire population model is used as a part of a more generic model. When the model is not dependent on covariates, the modeling problem is simplified and the usage non-parametric estimation is possible.

The Kaplan-Meier model was proposed by [6], who proved that it is the maximum likelihood estimator among the class of all possible functions for the survival function in an entire population model scenario. This estimator can be expressed as:

$$S^*(t) = \prod_{j \in \{1,\dots,N | (T_j \leq t),(E_j=1)\}} \frac{N-j}{N-j+1}, \tag{2.10}$$

where T_j is the minimum between the event time and the censoring time for subject j; T_j is sorted in an increasing order of T_j; and $E_j = \mathbb{1}(\text{event happened at } T_j)$ is an indicator function which is 1 if an event was observed for j and 1 if j was censored. The Kaplan-Meier model can also be used in competing risks scenarios as shown in [7].

The Nelson-Aalen model is an alternative approach proposed by [8]. Instead of performing maximum likelihood estimation, it models the cumulative hazard function following a martingale assumption. This estimator can be expressed as:

$$\Lambda^*(t) = \sum_{j \in \{1,\dots,N | (T_j \leq t)\}} \frac{E_j}{r_j}, \tag{2.11}$$

where r_j is the number of subjects that are not yet censored and have not yet experienced the studied event immediately before T_j.

The Aalen-Johansen [9] uses a martingale assumption to study the transition of states in a Markov chain and in a competing risks scenario can be used to estimate the cumulative incidence function:

$$F^*_{cause=k}(t) = \sum_{j \in \{1,\ldots,N | (T_j \leq t)\}} S^*(T_j^-) \frac{E_{j;k}}{r_j}, \tag{2.12}$$

where $S^*(T_j^-)$ is the survival function estimated with the Kaplan-Meier method at the time right before T_j; $E_{j;k}$ is 1 if event k happened at T_j and 0 otherwise; and r_j has the same meaning as in equation (2.11).

2.2.2 Proportional hazards model

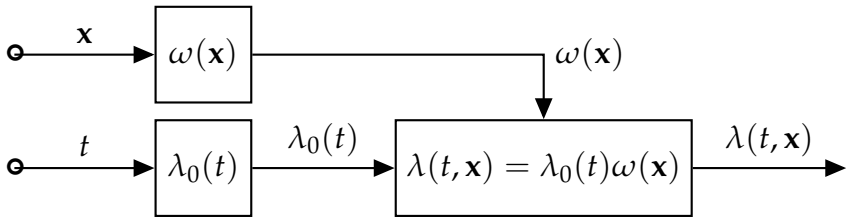


Figure 2.1.: Graphical description of the PH model.

The proportional hazards (PH) model expresses the survival probability as the combination of two separate components, one that depends on time and other that depends on the covariates. This is done through the expression $\lambda(t,\mathbf{x}) = \lambda_0(t)\omega(\mathbf{x})$, where $\lambda(t,\mathbf{x})$ is the hazards function, $\lambda_0(t)$ the baseline hazard function, and $\omega(\mathbf{x})$ the hazard ratio, which is strictly positive. Figure 2.1 provides a graphical description of a generic proportional hazards model. This model was originally proposed by [2] and its original formulation is known as the Cox model. In this formulation, the function $\omega(\mathbf{x})$ is given by a log-linear regression model: $\omega(\mathbf{x}) = \exp(\beta^T\mathbf{x})$, where $\beta \in \mathbb{R}^{N_x}$ are the coefficients in the regression model. It is shown in [2] that, if the baseline hazard function is strictly positive and increasing but unknown, a partial likelihood function can be defined. The original formulation of the partial likelihood is defined for the case in which $\omega(\mathbf{x}) = \exp(\beta^T\mathbf{x})$, but it can be extended for a generic $\omega(\mathbf{x})$ function:

$$\mathcal{L} = \prod_{j=1}^{N} \left[\frac{\omega(\mathbf{x}_j)}{\sum_{k=j}^{N} \omega(\mathbf{x}_k)} \right]. \tag{2.13}$$

The typical training procedure for a proportional hazards model is divided into two steps. First, $\omega(\mathbf{x})$ is determined through partial likelihood maximization. Second, $\lambda_0(t)$ is estimated maximizing the total likelihood with $\omega(\mathbf{x})$ fixed. There are several methods in the literature though which this estimation can be performed. One notable example is the method proposed by [10] in which the baseline hazard function is modeled with a piecewise linear function that has inflection points at the times in which an event has been observed. An analytical expression for the slope at each interval is provided through maximum likelihood estimation. Another notable example is the method proposed by [11] in Section 4.3, p. 84. It performs non-parametric maximum likelihood estimation of the baseline function given a fixed hazard ratio $\omega(\mathbf{x})$. The estimation procedure leads to a piecewise constant baseline hazard function and an analytical expression for it is provided. A competing risks extension of the proportional hazards models was proposed by [3], with the hypothesis that cause specific hazard function is proportional to a baseline hazard function. Another competing risks extension was provided by [4]. This model provides direct estimation of the cumulative incidence function: $F_j(t, \mathbf{x}) = 1 - \exp(-\int_0^t \lambda_{j0}(\nu) \exp(\beta^T \mathbf{x}(\nu)) \mathrm{d}\nu)$.

The hypothesis made in the Cox model limits the survival function that might be assigned to each set of input covariates in the following ways:

(i) The function $\omega(\mathbf{x})$ is restricted to a log linear model.

(ii) The influence exerted by the covariates in the survival function is not time dependent.

(iii) The optimization of function $\omega(\mathbf{x})$ is performed with the partial likelihood and the baseline hazard function is estimated with $\omega(\mathbf{x})$ frozen, which might lead to suboptimal results.

The restriction (i) has been dealt with in the literature by modeling $\omega(\mathbf{x})$ with a nonlinear model. The model proposed by [12] extends the Cox model modeling the hazard ratio $\omega(\mathbf{x})$ with a neural network. Initially, this has not resulted in any measurable improvement to the algorithm performance. However, later developments on neural networks made it possible to achieve better results than with the Cox model. The Faraggi & Simon model was extended in [13] with a deep neural network architecture; convolutional neural networks were used in [14] to provide an additional extension of the model in which it is possible to use information from images in the survival prediction. Both extensions of the Faraggi

& Simon model achieved better performance than the Cox model. A modified framework was used in [15], where the function $\omega(\mathbf{x})$ is computed with a neural network and an additional layer is used to estimate the survival function at a series of observation times. In this method, the function $\omega(\mathbf{x})$ is computed using the Cox partial likelihood, but it is not restricted to the proportional hazards assumption, since the link function that is used to predict the survival function might be changed. Nevertheless, this does not solve the restriction (ii), since there still must be a link between $\omega(\mathbf{x})$ and $S(t, \mathbf{x})$, and $\omega(\mathbf{x})$ is not time dependent. The Cox-Time model [16] was the first to solve this restriction by representing $\omega(\mathbf{x}, t)$ as the output of a neural network that has t as one of its inputs. The main limitation of this structure is that it makes the computational cost quadratic in the size of the dataset for both training and estimation. This creates the requirement for the use of approximations in the baseline hazard estimation and in the computation of survival function. For common healthcare applications where the input data is low-dimensional and a small neural network can be used, this approach is feasible despite the increased computational cost. Nonetheless, it would most likely not be possible to scale up this approach to more complex types of data as images or detailed health records where much larger models would be required.

The restriction (ii) has been often dealt with in the literature with solutions that are hand tailored to specific applications. For example, [17] modeled time dependent effects of covariates in kidney transplant patients with severe infections using natural cubic splines. Generic parametrizations for time dependency in proportional hazards models in which the coefficients $\beta(t)$ of the hazard ratio are made time dependent were proposed in several works that apply different types of splines to model these time dependencies [18–21]. Nevertheless, this approach does not generalize automatically to non-linear models, specially to neural networks, since the number of coefficients is much higher than in a linear model.

The restriction (iii) arises form the fact that the baseline hazards function is left completely unspecified when estimating $\omega(\mathbf{x})$. Nevertheless, there are in the literature representations for the baseline hazards function that solve this problem. The estimator proposed by [10] allows the total likelihood of the model to be obtained as a function of $\omega(\mathbf{x}) = \exp(\boldsymbol{\beta}^T \mathbf{x})$. Within the derivation of the non-parametric approach proposed by [11], it is mentioned during the derivation that Meier has proposed in 1978 in a personal communication to estimate the

parameters of both $\boldsymbol{\beta}$ and the baseline hazards function using the total likelihood function. Another notable formulation in which this restriction is solved is the flexible parametric (FP) model [22]. It models the cumulative baseline hazard function with natural cubic splines and performs joint optimization of $\Lambda_0(t)$ and $\omega(\mathbf{x})$. The FP model is built within a more general framework than the proportional hazards. The mathematical formulation for it is: $g[S(t;\mathbf{x})] = \log(\Lambda_0(t)) + \beta\mathbf{x}$, where $\Lambda_0(t)$ is modeled with natural cubic splines, and $g_\theta(S) = \log((S^{-\theta} - 1)/\theta)$ is a generic link function. If $\theta \to 0$, the model is reduced to the proportional hazards assumption. Nevertheless, similarly to the model proposed by [15] it is not a general model, since it still require that there is some type of correspondence between the shapes of the survival function for different subjects. A competing risks extension of the FP method was proposed by [23].

2.2.3 Accelerated failure time model

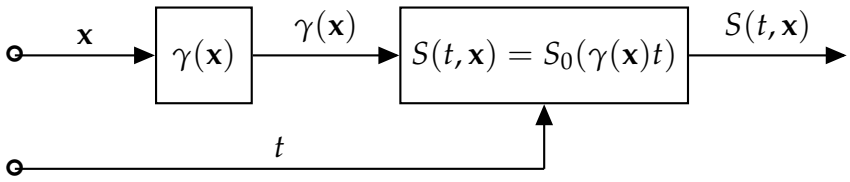


Figure 2.2.: Graphical description of the AFT model.

The accelerated failure time (AFT) model was proposed by [2] to provide a counterpoint to the Cox model. It is based on the hypothesis that the covariates influence the failure process by accelerating or decelerating it. Figure 2.2 provides a graphical representation of this model. In it, a function of the covariates $\gamma(\mathbf{x})$ is computed and multiplied by the time to provide the input for the baseline survival function S_0. Similarly to the proportional hazards model, the original formulation also follows a log-linear model $\gamma(\mathbf{x}) = \exp(\beta^T\mathbf{x})$. In the original formulation of the AFT model, no method is provided for estimating the parameters in it. A training method based on least squares estimation of the logarithm of the survival time was preposed in [24], with a Kaplan-Meier estimator of the prediction error being employed to weight the squared errors of each uncensored prediction. The main weakness of this approach is that it assumes that for any given $\mathbf{x}$ the event time is a random variable with lognormal probability distribution, which is often not the case.

An alternative estimation approach within the AFT model was proposed by

[25]. In this approach, the estimation is based on quantile regression instead of maximum likelihood estimation. This means that instead of estimating the expected value of the logarithm of the time to event, this approach estimates the median of the logarithm of the time to event. This overcomes the restriction of the survival function to a lognormal distribution, but presents two important limitations:

(i) The shape of the survival function is influenced by the covariates only through a time scale factor.

(ii) The function $\gamma(\mathbf{x})$ is restricted to a log linear model.

The limitation (i) was solved by [26]. This was done with the inclusion of quantile dependent coefficients in the quantile regression problem. This means that, for each subject, the model will predict the log of the time at which the probability of the event to have already occurred is equal to τ. This estimation is made through linear model in the form $\beta(\tau)^T\mathbf{x}$ for $0 < \tau < 1$, where the coefficients depend on τ. More specifically, the coefficients $\beta(\tau)$ are modeled as piecewise-constant functions. The estimation of the coefficients are based on martingale theory and are performed through the minimization of an objective function. Proofs of uniform consistency and weak convergence are provided.

The limitation (ii) is more difficult to overcome in the context of AFT models than in proportional hazards models. A neural network version of the AFT model was proposed in [27]. This method overcomes the linearity restriction in the original model, but it is based on maximum likelihood estimation similarly to [24] and is consequently restricted to a lognormal probability distribution. A support vector regression model for survival analysis was proposed by [28] within a framework similar to [24], in which an estimation for the time to event is provided, but it is not associated with a probability distribution of the error of this estimation, which means that the model does not completely specify the survival function. A neural network extension of the method proposed in [26] is not trivial since the objective function used is not continuous, which makes it computationally expensive to be optimized for a nonlinear model.

2.2.4 Aalen additive model

The Aalen additive model [29] is a covariate dependent extension of the Nelson-Aalen model described in Section 2.2.1. In this method, the estimation of a piecewise constant cumulative hazard function is performed similarly to the Nelson-Aalen model, but each step is computed through the solution of a linear algebra problem. The hazards function in this model is:

$$\lambda(t, \mathbf{x}) = \mathbf{x}(t)\boldsymbol{\omega}(t)\,, \tag{2.14}$$

where $\boldsymbol{\omega}(t)$ are time dependent coefficients and the covariates $\mathbf{x}(t)$ might vary with time. The integral of the coefficients is defined by $\mathbf{A}(t) = \int_0^t \boldsymbol{\omega}(t)\mathrm{d}t$, where $\boldsymbol{\omega}(t)$ is a vectorial function of time, and it is estimated by:

$$\mathbf{A}(t) = \sum_{T_k \leq t} [\mathbf{x}(T_k)^T \mathbf{x}(T_k)]^{-1} \mathbf{x}_k E_k\,, \tag{2.15}$$

where $\mathbf{x}_k$ are the covariates of a subject in the training set, E_k indicates if T_k is an event or a censoring time and $\mathbf{x}(t)$ is a matrix where row k is $\mathbf{x}_k$ if $T_k \leq t$ and zero otherwise. One limitation of the Aalen additive model is that it provides no guarantee that the cumulative function is positive. Although the estimation framework used to derive this model is difficult to generalize to a nonlinear machine learning approach, the idea of splitting the estimation into different models can be more easily generalized to machine learning methods. Indeed, it is possible to see the discrete time models in Section 2.3.2 as machine learning successors of the Aalen additive model.

2.2.5 Extended Hazard Model

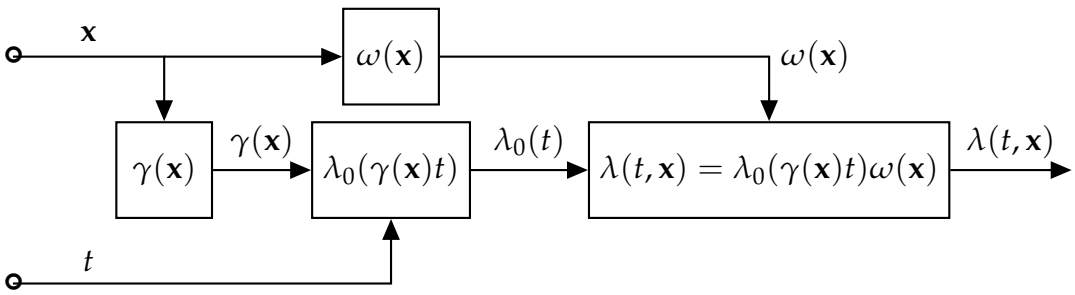


Figure 2.3.: Graphical description of the EH model.

The extended hazard (EH) model [30] provides an extension to previous methods in the form of a combination of the proportional hazards and accelerated failure time model. There, the input variables $\mathbf{x}$ can influence both the gain and time scale of the hazard function in the form $\lambda(t, \mathbf{x}) = \lambda_0(t\gamma(\mathbf{x}))\omega(\mathbf{x})$. In the original formulation, both γ and ω are given by log-linear models, in the form: $\gamma(\mathbf{x}) = \exp(\beta_1^T\mathbf{x})$ and $\omega(\mathbf{x}) = \exp(\beta_2^T\mathbf{x})$. A neural network extension of the EH model has been proposed in [31]. Despite being more generic than the PH or AFT models, the EH still suffers from the same type of limitation as their less generic antecedents where the estimated function is constrained to a particular family of function. The type of extension provided by the EH model allows a smooth combination of both families but is not enough to allow the representation of any arbitrary pattern that might be encountered in the data.

2.2.6 Probability score models

Section 2.5 deals with metrics that are used to evaluate survival models. Some attempts have been made in the literature to perform inference using those metrics instead of performing maximum likelihood estimation or other traditional statistical estimation approaches. It was shown in [32] that the Cox partial likelihood is a lower bound to the concordance index [33], which will be further described in Section 2.5.1. Since the concordance index is not a continuous function, the direct optimization of it is computationally expensive. As an alternative the article proposes optimization of the concordance index based on other lower bounds to it. The results achieved are comparable to results of the Cox model. However, the proposed model does not provide an estimation of the survival function. Instead, it provides a survival ranking score for subjects in a population. Similar methods were proposed using support vector machines to estimate the ranking between survival times in which the concordance index is optimized either using a lower bound [34] or directly [35].

A method in which the estimation is performed using the integrated Brier score was proposed in [36]. The integrated Brier score is considered a proper score rule because it is guaranteed to have maximum expected value for the true survival probability distribution. In this model, the parameters of a log-normal probability distribution are fitted as outputs of a neural network and optimized according to this metric. The model has the advantage of not suffering from the limitations in the proportional hazards model. Indeed, all the parameters in the probability

distribution can be determined as the output of an arbitrary non-linear function and the entire model is optimized with a single metric, which prevents it from suffering from sub-optimality. The experimental results showed that a better result was achieved when the parameters were estimated through the minimization of the integrated Brier score as opposed to maximum likelihood estimation of the same model. Nevertheless, the probability distribution is restricted to a lognormal distribution that might not capture the true probability distribution for the time to event.

2.3 Divide and conquer models for survival analysis

While the search space for survival probability distributions might have infinite dimension and the survival data used to estimate the model is unidimensional, the dataset in some applications is so large that this problem can be neglected. This section is devoted to the description of approaches in which the infinite dimension search problem is divided into a large number of simpler problems and only a small part of the data is relevant for solving each problem. The brute force methods described in the present section make less efficient use of the data than the methods described in Section 2.2 in the sense that patterns that are common to the entire population have to be learned multiple times making use of subsets of the data, which requires a larger amount of data then would be required if these patterns could be learned using the entire dataset. However, they have the advantage of being able to represent any arbitrary survival function and, if the amount of data available is enough, they can be used to represent patterns that cannot be represented by other methods.

There are two main approaches for developing a brute force model for survival analysis. First, dividing the data in groups with similar covariates and using entire population models for each group. Second, transforming the survival function into a discrete time probability distribution with a large number of time steps so that the problem at each step is reduced to a classification problem. The first approach is described in Section 2.3.1 and the second is described in Section 2.3.2.

2.3.1 Nearest neighbour models

Nearest neighbor models for survival analysis use the idea of dividing the subjects into groups with similar behavior and using an entire population model for each group of individuals. Two methods for dividing the subjects were proposed in [37]: selecting the k nearest neighbors to a subject; or assigning different weights for the subjects according to a kernel function that measures how similar two subjects are. In the first case, a Kaplan-Meier estimation is made using only subjects that are close to the target subject. In the second case, Kaplan-Meier is adapted using weights for each subject that reflect how close their covariates are from the target subject. Both approaches suffer from the fact that the division between subject groups is not influenced by how relevant each covariate is to the survival function.

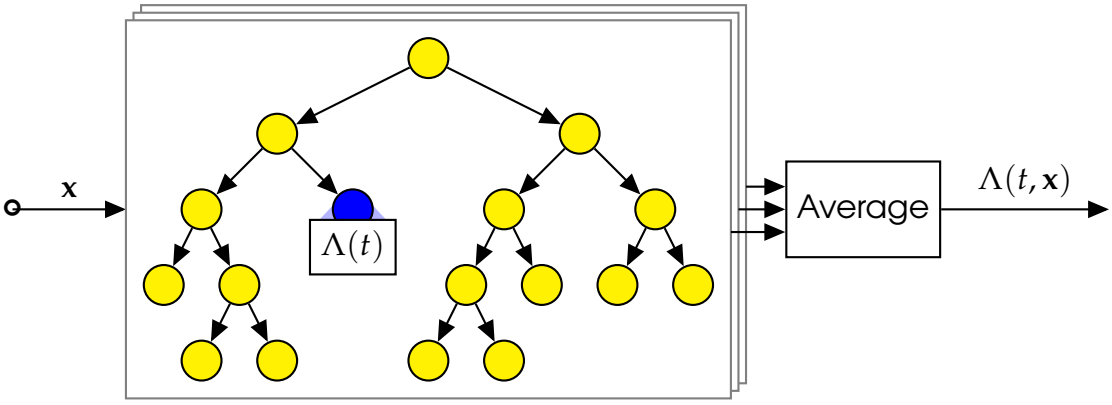


Figure 2.4.: Graphical description of the random survival forest model.

This limitation is overcame with the use of random survival forest, which is an adaptation of the random forests model to survival analysis problems. The first random survival forest method was proposed by [38]. A graphical representation of a random survival forest is provided in figure 2.4. This methods creates a set of binary trees that divide the subjects according to their covariates in a way that maximizes the difference in survival. For each binary tree, a random subset of the database is selected. The training algorithm for each binary tree begins with all the subjects in the same leaf and recursively find a covariate and a threshold to divide the subjects in two child leafs with the greatest possible survival difference among them. This procedure is performed until a minimum amount of event samples is left in each leaf. After each tree has been created, each subject will be associated to a single leaf node within a tree, where an estimation of the cumulative hazard function is obtained using the Nelson-Aalen

method described in Section 2.2.1. The final estimation for a given subject is the average over all trees of the cumulative hazard function. The random survival forest algorithm was extended to the competing risks scenario in [39]. In this approach, the cumulative incidence function (CIF) is estimated with the Aalen-Johansen model. The final estimation for a given subject is obtained by the average of the CIF estimation for the same subject in each tree.

In classification problems, random forests are known for producing biased results for underrepresented samples. Based on this fact, [40] proposes an adaptation of boosting algorithm for survival analysis in which each survival tree in the forest receives a different weight based on the classification accuracy. The model is built in a competing risks framework, and the Aalen-Johansen estimator is used at each leaf node.

Although no proof of consistency is provided in the original derivation of the random survival forest algorithm, [41] provides a non-asymptotic upper bound for the error for a broad class of nearest neighbor type algorithms, including the two approaches proposed by [37] and a slightly modified version of the random survival forest [39] in which the average of the Kaplan-Meier estimators in each leaf is used instead of the average of the Nelson-Aalen estimators. Additionally, a new version of the random survival forest is proposed in which, instead of using the Kaplan-Meier estimation at each leaf, a kernel function is constructed which corresponds to the probability of two subjects to be in the same leaf in a survival tree. A weighted version of the Kaplan-Meier model is applied making use of that kernel to compute the weights. Experimental results show that this new version of the random survival forest algorithm has better performance than the other algorithms considered in the study.

2.3.2 Discrete time models

The discrete time models for survival analysis simplifies the survival modeling problem by dividing the time interval in which the model is analysed into a quantized sequence of times and reframing the problem into a composition of a different classification problem for each time step. This idea can be traced from Aalen additive model [29] described in Section 2.2.4, where the hazard function is computed as a linear function of the input variables where the coefficients vary with time. A discrete time model named multi-task logistic regression

(MTLR) was proposed in [42], where the survival function for each time point is represented by a logistic regression model. A comparison is made with classical survival analysis models, but no quantitative comparison was made in terms of conventional survival analysis metrics. The MTLR model was extended in [43] making use of state of the art machine learning tools. A different approach was proposed in [44] where the survival function at each point is predicted with a linear model without use of logistic activation function and a cost function based on the Frobenius norm of the survival status error is used instead of a likelihood function. The MTLR model was extended in [45] through the definition of a different cost function that makes the model compatible with competing risks. In this formulation, the model predicts the incidence function at each time step for each possible event as outputs of a neural network. An extension of the method proposed in [45] was made [46], where the output of a recurrent neural network (RNN) is used to predict the instantaneous hazard rate at each time step. Other methods for applying RNN to discrete time survival models were proposed by [47] and [48], in which the objective function optimized to train the RNN is based on the average of the cross entropy across over all sampling times. While in a classification problem the cross entropy is equivalent to the likelihood, it is not the case in these models since it would only be possible to average the likelihood over all the sampling times if the survival function at each time was independent from each other and this is not the case.

2.4 Monte Carlo models for survival analysis

The models in Sections 2.2 and 2.3 provide explicit estimations of the survival distribution or the survival time for any subject. However, there are models that are intrinsically generative because they do not provide an explicit representation of the survival probability distribution and requires Monte Carlo sampling to estimate it. Most of these model are built within a Bayesian framework, using a parametric or semiparametric model for the survival probability distribution and sampling the parameters of the model through Bayesian inference. Nevertheless, this is not always the case and Section 2.4.4 presents a generative model for survival analysis that is based on generative adversarial networks instead of Beyesian inference.

2.4.1 Gaussian processes

Gaussian processes are extensions of the multivariate Gaussian distribution to infinite dimension spaces. Gaussian processes have been used extensively in the literature to perform probabilistic estimations of real valued functions. For that purpose an "a priori" distribution is assumed for the function that will be estimated. In this "a priori" distribution, a kernel function is chosen to express the correlation between values in different evaluation points of the function. According to the kernel function, a different class of "a priori" distribution will be generated. The training data is then used to perform Bayesian estimation, making it possible to know the "a posteriori" distribution. This provides, for each input value, a probability distribution of the target value. In cases in which the likelihood of a dataset given the target function is not Gaussian, it is usually not possible to perform analytical inference of the "a posteriori" distribution. Nevertheless, it is possible to sample from that distribution with Markov Chain Monte Carlo (MCMC) estimation. Alternatively, it is possible to compute an approximation of the "a posteriori" distribution. The first attempts in the literature to apply Bayesian inference for survival analysis with stochastic processes relied on other stochastic processes instead of Gaussian processes. A notable example is provided by [49] in which a gamma process is used. In recent works, the focus has been shifted to Gaussian processes models, which are broadly used in the machine learning literature.

The application of Gaussian processes to survival analysis requires the computation of the likelihood of a data sample given the output of the Gaussian process. The methods available in the literature perform this task by using simple parametric survival models and consider the parameters to be outputs of the Gaussian process. The usage of Gaussian processes to estimate a piecewise constant baseline function in a Cox model was proposed by [50]. This idea was extended by [51] estimating jointly the covariates coefficients and the parameters in the piecewise constant baseline hazards function. Alternatively, [52] uses Gaussian processes to estimate the parameters in an accelerated failure time model in a competing risks scenario. Chained Gaussian processes are used for survival modeling in [53]. This method allows parameters of the survival model to be dependent on the covariates even if they cannot be framed in a generalized linear model. The usage of a Gaussian process to estimate a time dependent hazard ratio in the proportional hazards framework was proposed by [54]. This allows for the covariates to have non-linear time dependent effects on the sur-

vival function. However, the model does not allow the explicit computation of the survival function. Instead, it allows sampling from the survival time probability distribution. A deep Gaussian model was used by [55] to estimate the survival time in terms of the covariates within a competing risks framework. Survival times cannot generally be modeled directly as outputs of a Gaussian process since it is wrong to suppose that the survival time follows a Gaussian distribution. With deep Gaussian models, latent variables are added to the model and the target value is modeled as the output of a series of Gaussian processes where the input of each model is the output of the previous model. As a result, the target variable probability distribution is no longer Gaussian, which makes it possible to model the survival time directly as the output of this model.

2.4.2 Deep exponential families

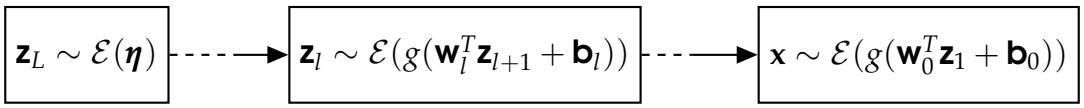


Figure 2.5.: Graphical description of the deep exponential family model.

Generative survival models have also been proposed in the literature using deep exponential families [56], which are generative models that use a composition of samples from exponential family probability distributions to generate samples from a generic multivariate probability distribution. Figure 2.5 gives a graphical representation of a deep exponential families model. The sampling process in this model is performed as follows: first, a vector $\mathbf{z}_L \in \mathbb{R}^{N_L}$ is sampled from an exponential family $\mathcal{E}$ with parameters $\boldsymbol{\eta} \in \mathbb{R}^{N_L \times 2}$; then, each intermediate layer $\mathbf{z}_l \in \mathbb{R}^{N_l}$ is sampled from an exponential family $\mathcal{E}$ with parameters $g(\mathbf{w}_l^T \mathbf{z}_{l+1} + \mathbf{b}_l)$, where g is a nonlinear link function and $\mathbf{w}_l \in \mathbb{R}^{N_l \times 2 \times N_{l+1}}$ and $\mathbf{b}_l \in \mathbb{R}^{N_l \times 2}$ are respectively weights and biases; finally, the visible variables $\mathbf{x} \in \mathbb{R}^{N_x}$ are sampled through the same process as the intermediary layers $\mathbf{z}_l$. Once the model has been trained, it is possible to sample part of the visible variables conditioned to a subset of them. Training in this model is performed through variational inference, using the "black box" algorithm [57]. Variational inference is an approximate training method that is used to train Bayesian methods in which it is not possible to compute the likelihood analytically. The method relies in providing an analytical lower bound to the likelihood and optimizing it instead of the likelihood. While variational inference requires that a lower bound is found for

each model that is going to be optimized, the "black box" algorithm provides a general variational bound for a class of models.

The first deep exponential family application to survival analysis was proposed by [58]. In this model, the covariates $\mathbf{x}$ are modeled as the visible layer from a deep exponential family and the event time follows a Weibull distribution in which $\lambda = \log(1 + \exp(\mathbf{z}_1^T \mathbf{a} + b))$ and k is fixed, where $\mathbf{z}_1$ comes from the deep exponential family and the values $\mathbf{a}$ and b are sampled from Gaussian distributions: $\mathbf{a} \sim \text{Normal}(0, \sigma_W)$; and $b \sim \text{Normal}(0, \sigma_b)$. The survival times can be sampled from the deep exponential family conditioned to the covariates. A further development to this method have been made by [59]. In this work, the Weibull distribution for the event time is expanded by applying a sequence of monotonic transformations to t. This allows for more flexible representations of the probability distribution and the experiments show that this results in an better survival model.

2.4.3 Lomax delegate races

Lomax delegate races have also been used in generative survival models. In [60], a framework has been proposed in which failure is modeled as a race between an infinite number of agents and the event time is considered to be the minimum between the event time in all those agents. Each agent is modelled as following a Lomax distribution, which is the result of an exponential distribution $t \sim \text{Exp}(\lambda)$ where the parameter λ is sampled from a Gamma distribution $\lambda \sim \text{Gamma}(r, 1/b)$. A covariate dependent version of this distribution is defined by making $1/b = e^{\beta^T \mathbf{x}}$. The number of agents is limited to K in order to make the estimation computationally tractable. Additionally, in scenarios in which it is necessary to model competing risks, the agents are divided into J groups with K agents which so that a particular event is considered to have occurred if an agent in the correspondent team has had the minimum event time. Bayesian inference in this model is performed through Gibbs sampling.

2.4.4 Generative adversarial networks

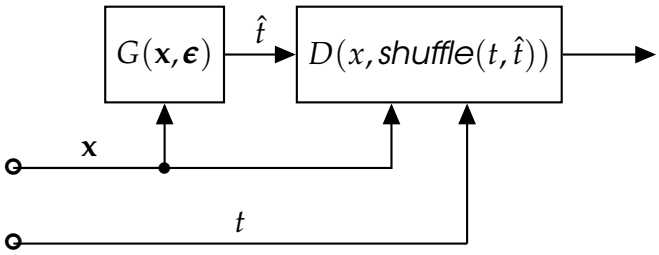


Figure 2.6.: Graphical description of the deep adversarial time-to-event (DATE) model.

Generative adversarial networks (GANs) have been proposed by [61] and are pairs of neural networks that model the underlying probability distribution for a given set of covariates. It is composed of two neural networks: a generator and a discriminator. The goal of the generator is to generate random samples from the covariates distribution and the goal of the discriminator is to correctly identify if a sample comes from the real dataset or is artificial. A survival analysis version of generative adversarial networks (GANs) has been proposed by [27]. Figure 2.6 gives a graphical description of it. In this model, the generator $G(\mathbf{x}, \boldsymbol{\epsilon})$ predicts a survival time given the covariates $\mathbf{x} \in \mathbb{R}^{N_x}$ and a vector of normally distributed random variables $\boldsymbol{\epsilon} \in \mathbb{R}^{N_E}$. The discriminator $D(x, shuffle(t, \hat{t}))$ estimates whether or not, in a pair (x, t), the time comes from a true sample or from the generator. Ideally, this should allow for the generator to learn a good representation of survival time in terms of the covariates. However, the structure of the generator model favors the distribution of the generated time to be close to Gaussian, which is typically not the case in survival scenarios.

2.5 Model evaluation

The study of metrics to evaluate survival models an important part of survival analysis. The recent developments in machine learning techniques for survival analysis modeling have an impact to evaluation metrics, since some classical evaluation metrics rely on assumptions that are not valid for all models.

2.5.1 Concordance index

The most used metric for evaluating survival models is the concordance index [33]. It is an adaptation of the area under receiver operation characteristic curve (AUROC) and it evaluates how well a survival model distinguishes the survival time in a pair of subjects based on their covariates. This is done by evaluating all the possible pairs of two subjects and selecting the pairs that can be ordered. This is the case when neither of the subjects has been censored or when only one subject has been censored but the censoring time is greater or equal to the event time of the other subject. In the later case, the pair can be ordered because the event time for the censored subject is certainly greater than the event time for the other subject. The concordance index is given by:

$$CI = \frac{1}{|\epsilon|} \sum_{\epsilon_{ij}} 1_{f(\mathbf{x}_i) < f(\mathbf{x}_j)}\,, \tag{2.16}$$

where $|\epsilon|$ is the number of comparable pairs and $1_{f(\mathbf{x}_i)<f(\mathbf{x}_j)}$ is an indicator function that is: 1 when the predictor $f(\mathbf{x})$ correctly orders the subjects; and 0 when it orders them incorrectly. The values form the concordance index can vary from 0 to 1. A value of 0.5 means that the model does not provide any information on the ordering of the events. A value of 1 means that the model is a perfect predictor of the ordering of the events. A value of 0 means that the model always makes the wrong order prediction.

The censoring in the data might cause the concordance index to be biased, since the result of the concordance index in a censored dataset might be different from what it would be in case the same dataset was not censored. This happens because subjects with larger event times have a greater chance of being censored before the event is observed, and as a result the distribution of event times in the uncensored dataset is different from what it would be if no subject had been censored. A modification to the concordance index that prevents the measurement to be susceptible to the censoring probability was proposed by [62]. With this modification, all pairs in the concordance index computation have a weight inversely proportional to the square of the non-censoring probability at the time of the first event in the pair. The non-censoring probability is computed through the Kaplan-Meier method, but swapping events with censoring. This estimator has been generalized by [63] through the inverse of the probability of censoring weighted (IPCW) method, which takes into account scenarios in which the censoring probabilities depends on the covariates. In

this method, each pair has a weight that is inversely proportional to the product of the non-censoring probability for each subject at the time of the first event in the pair. The concordance index computed with the IPCW method for a given time t is given by:

$$CI_{IPCW} = \frac{\frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\mathbb{1}\{S(t,\mathbf{x}_i) > S(t,\mathbf{x}_j)\}\mathbb{1}\{T_i < T_j\}N_i(t)}{\frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\mathbb{1}\{T_i < T_j\}N_i(t)}, \tag{2.17}$$

where the $N_i(t)$ is the event indicator for subject i at time t. A competing risks extension of the IPCW estimation of the concordance index was proposed by [64].

2.5.2 Time dependent concordance index

Although the IPCW method is able to estimate a concordance index as a function of time, [65] showed that there might not be a one to one correspondence when ranking subjects according to the survival function and the expected value of the event time. This means that, in these scenarios, the concordance index can not be justified in terms of the AUROC. As an alternative, they have proposed the time dependent concordance index, defined as a weighted average of the AUROC estimate at each time:

$$C^{td} = \frac{\sum_{k=0}^{K} AUROC(t_k)w(t_k)}{\sum_{k=0}^{K} w(t_k)}, \tag{2.18}$$

where:

$$AUROC(t_k) = Pr\{S(t_k,\mathbf{x}_i) < S(t_k,\mathbf{x}_j)|N_i(t_k); N_j(t_k)\}. \tag{2.19}$$

2.5.3 Brier score

The Brier score [66] is an important metric in classification problems and a time dependent version of that metric was proposed by [67], which is called weighted residual sums of squares (WRSS). Similarly to the concordance index, this metric is biased in the presence of censoring. An unbiased estimator of the Brier score for survival analysis scenarios with censoring was proposed by [68]. This estimator relies on the hypothesis that the censoring probability does not dependent on the covariates. A generalization of this estimation with covariate

dependent censoring was proposed by [69]. In this generalized formulation, the estimator is:

$$WRSS(t, S, G) = \frac{1}{n}\sum_{i=1}^{n}[\mathbb{1}(T_i > t) - S(t, \mathbf{x}_i)]^2 \left[\frac{\mathbb{1}(T_i \leq t)E_i}{G(T_i^-|\mathbf{x}_i)} + \frac{\mathbb{1}(T_i > t)}{G(t|\mathbf{x}_i)}\right] . \quad (2.20)$$

2.5.4 Calibration plots

Even when a survival model presents better concordance index and Brier score relative to other models, it is possible for the model to produce undesired patterns in the probability distribution estimation. In order to avoid this situation, it is customary to make calibration plots of survival models to guarantee that the average survival prediction for a population is consistent. This can be done either for the entire population or for subsets of it. In the first case, the survival function is estimated for each subject in the population and the average of the survival estimations is compared with the Kaplan-Meier estimation for that population. In the later case, the same procedure is followed, but for subsets of the population. The subset of the population are separated either using the quantiles of the survival prediction in a given point for that model or by grouping subjects with similar values for a covariate of interest. Examples of the usage of calibration plots can be found in [2] and [4].

3

Metaparametric neural networks: a generic hierarchical modelling framework

Neural networks are parametric functions that can be used to approximate almost any continuous multivariate function. However, this versatility comes at the cost of obscuring the relationship between the input variables and the output. This "black box" structure is an important obstacle to the larger scale application of neural networks. In healthcare for example, identifying the underlying mechanism producing an observed health outcome may be the goal of the predictive model. A key function of the applied statistical model is thus to explain the contributory effects of different input variables upon the outcome. A further limitation of this "black box" structure is the inability to validate the model. Whilst sensitivity analyses can show trends within the data, the error bounds generated cannot be relied upon throughout the input space. Errors may be larger in the unseen data than those inferred from sensitivity analysis in parts of the sparsely sampled input space. Nonetheless, neural networks often perform considerably better than more easily interpretable statistical models, having achieved demonstrable success in several domains.

In the present chapter, we propose the metaparametric neural network

(MNN), a novel framework with better interpretability than "black-box" neural networks while still being able to represent arbitrary functional relationships. Indeed, we prove in Theorem 1 the universal approximation property for MNNs showing that they are able to approximate any continuous function arbitrarily well. This is achieved through a "grey-box" hierarchical structure where some of the input variables have a more explicit representation than others. As a result of their universal approximation property, MNNs can be applied to any problem where a multi-layer perceptron neural network could be applied, achieving a more intelligible representation for a subset of the input variables. In particular, this can be applied to survival analysis where estimation requires an explicit representation of how the time influences the event probability, as shown in Chapter 4. This requirement is the major obstacle for making completely generic neural network extensions of survival models like it has been achieved in other domains. In regression and classification problems for example, the Universal Approximation Theorem 1 shows that a neural network can fit any continuous function arbitrarily well as long as the number of parameters in the model is large enough. This property enables their application to virtually any task with a guarantee that the model is generic enough to represent the patterns present in the data. Survival models require that the target function is monotonic over time (i.e. if the target function is the survival function it can not increase with time and if it is the cumulative hazard function it can not decrease with time), and include codependency between time and other inputs. This hinders the direct application of neural networks to survival modeling. This requires either adaptations of the target function like in DeepHit [45] where the event probability distribution is discretized in time, or making the estimation extremely inefficient like in Cox-Time [16]. The hierarchical structure of MNNs can solve this problem as will be shown in Chapter 4.

3.1 Metaparametric neural network framework

The metaparametric neural network (MNN) framework is based on a hybrid association between neural networks and other parametric approaches with more explicit and intelligible outcomes. In particular, we show how neural networks can be integrated with basis functions and mixture models in order to achieve a hierarchical model that is more intelligible than standard neural networks without having to impose any restriction to the type of functions being

modeled. Basis functions and mixture models by themselves are capable of representing virtually any functional relationship, similarly to neural networks. When compared to neural networks, they have the advantage of allowing a better interpretability of the resultant model. This is the case of audio processing, which could be viewed as an estimation problem with only one input dimension given by time with a dataset where one target value is associated with each possible value for the input. A similar scenario is present in image and video processing, where the number of input dimensions is two and three respectively. In those applications, Fourier domain analysis has been shown to be of great value in both understanding the structure of the data and representing it effectively. However, the number of parameters in the model must grow exponentially with the number of input variables in order to keep the model generic and this is why they are often only used in problems with small number of input variables. With the MNN framework, it is possible to divide input variables so that most of them will be accounted for in a black-box neural network while a limited subset of them are represented with a more intelligible parametric representation.

The definition of an MNN relies on the definition of a neural network as it is one of the blocks of the MNN. Although several neural networks exist and can be employed in MNN models, in the present chapter we always assume a multilayer perceptron (MLP) architecture to focus on the higher level aspects of the MNN structure. An MNN can be defined as a model that computes $\boldsymbol{y} \in \mathbb{R}^{N_y}$ given $\boldsymbol{x} \in \mathbb{R}^{N_x}$ in the form:

$$\boldsymbol{h}_1 = a(\boldsymbol{W}_m \boldsymbol{x} + \boldsymbol{b}_m)\,, \tag{3.1}$$

$$\boldsymbol{h}_m = a(\boldsymbol{W}_m \boldsymbol{h}_{m-1} + \boldsymbol{b}_m)\,, \tag{3.2}$$

$$\hat{\boldsymbol{y}} = a(\boldsymbol{W}_M \boldsymbol{h}_{M-1} + \boldsymbol{b}_M)\,, \tag{3.3}$$

where W_m are matrices of parameters, b_m are vectors of parameters and $a(\cdot) : \mathbb{R} \to \mathbb{R}$ is an activation function, than can in principle be any function chosen when defining the model and it is aplyed element-wise in vectors. For simplicity, we denote the vectorized set of all parameters in the MLP as $\boldsymbol{\theta}$ in the form $\hat{\boldsymbol{y}} = \psi(\boldsymbol{x}; \boldsymbol{\theta})$.

Definition 1. *Let $\boldsymbol{\psi}(\boldsymbol{x}, \boldsymbol{\theta})$ be a parametric neural network with input variables $\boldsymbol{x} \in \mathbb{R}^{N_x}$ and parameters $\boldsymbol{\theta} \in \mathbb{R}^{N_\theta}$ and let $g(\boldsymbol{y}, \boldsymbol{\psi})$ be a parametric function of $\boldsymbol{y} \in \mathbb{R}^{N_y}$ with parameters $\boldsymbol{\psi} \in \mathbb{R}^{N_\psi}$, where $\boldsymbol{y}$ is a set of input variables disjoint from $\boldsymbol{x}$. We define the metaparametric neural network (MNN) $g(\boldsymbol{y}, \boldsymbol{\psi}(\boldsymbol{x}, \boldsymbol{\theta}))$ where the output of $\boldsymbol{\psi}(\cdot)$ serves as the parameters of $g(\cdot)$. This is a hierarchical model where the input variables are grouped into a set of implicit variables $\boldsymbol{x}$ and another set of explicit variables $\boldsymbol{y}$ that allows the outcome $g(\cdot)$ to be explicitly represented as a function of $\boldsymbol{y}$ for any particular value of $\boldsymbol{x}$.*

This definition was reproduced with permission from IEEE, [70] © 2021 IEEE. From the definition, it is possible to notice that the parameters $\boldsymbol{\psi}$ in function $g(\boldsymbol{y}, \boldsymbol{\psi})$ depend on variables $\boldsymbol{x}$. Therefore, the parameters are adapted for each value of $\boldsymbol{x}$ and this is property allows MNN models to approximate functions in the form $h(\boldsymbol{x}, \boldsymbol{y})$. In practice, this means that the values of $\boldsymbol{\psi}$ are not directly estimated, since in the overall model they do not act as parameters, but as intermediary values that are computed by the model with the use of $\boldsymbol{x}$ and $\boldsymbol{\theta}$. Indeed, in its complete formulation, the MNN equation is given by $g(\boldsymbol{y}, \boldsymbol{\psi}(\boldsymbol{x}, \boldsymbol{\theta}))$, which is a parametric function of $\boldsymbol{x}$ and $\boldsymbol{y}$ with parameters $\boldsymbol{\theta}$. Thus, the parameters set $\boldsymbol{\theta}$ suffice to specify the MNN and estimation can be performed similarly to any other neural network model by estimating only parameters $\boldsymbol{\theta}$.

The structure of a MNN model is represented in Figure 3.1, which was reproduced with permission from IEEE.

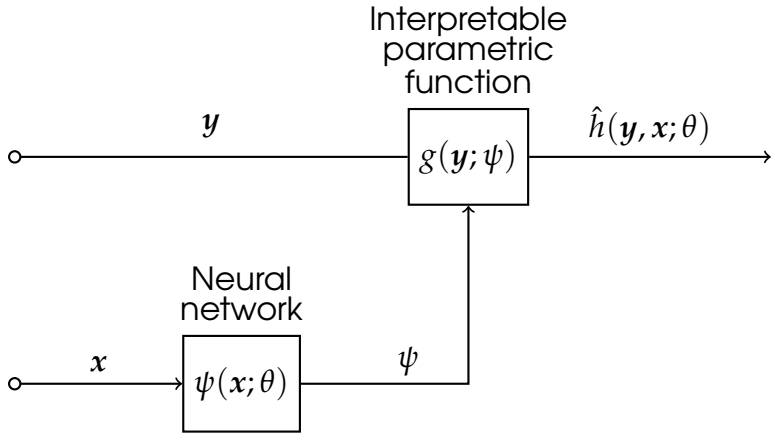


Figure 3.1.: Graphical description of a metaparametric neural network. [70] © 2021 IEEE.

3.2 Universal approximation theorem of metaparametric neural networks

In this section we provide a formal proof that, subject to specified constraints, the MNN will be capable of representing any continuous functions over a compact and convex space with an arbitrarily small error. This is done in Theorem 1, whose proof uses the results in Lemmas 1 and 2.

3.2.1 Introduction to Universal Approximation

The purpose of a universal approximation theorem is to guarantee that a particular class of models can approximate the target function so that the model choice does not impose any restriction to the estimation accuracy. To better illustrate the importance of this property, we can consider an example of a class of functions that does not have it. In the case of estimating the function

$$h(x_1, x_2) = x_1 + x_2 \tag{3.4}$$

using a class of parametric functions in the form:

$$g(x_1, x_2, \Theta, \Phi) = f(x_1, \Theta) g(x_2, \Phi)\,, \tag{3.5}$$

it would be impossible to achieve an accurate representation of the target function $h(x_1, x_2)$. This means that this particular choice of parametrization imposes a hypothesis to the function that is being modeled and if it is used to estimate a function that does not follow that hypothesis, the result will always be wrong.

The universal approximation theorem guarantees that it is theoretically possible with a given class of parametric functions to achieve an arbitrarily small error when estimating any continuous function in a particular domain. In practice, it is not possible to achieve an arbitrarily small estimation error because there are other factors limiting the estimation accuracy, including the size of the dataset used for estimation and the computational costs of estimating the model. Nonetheless, the theorem has the important role of guaranteeing that the choice of the class of functions will not impose any hypothesis that is incompatible with the function being modeled.

More specifically, a universal approximation theorem is formulated as follows.

It is desired to approximate a function $h(\mathbf{z}) : \mathcal{Z} \to \mathbb{R}^p$, where $\mathcal{Z} \subseteq \mathbb{R}^n$. For that purpose, a class of parametric functions is used where for every possible number of parameters m, a different version of the parametric function is provided $g_m(\mathbf{z}; \theta^{(m)}) : \mathcal{Z} \times \mathcal{T}^{(m)} \to \mathbb{R}^p$, where $\mathcal{T}^{(m)} \subseteq \mathbb{R}^m$. Since function $h(\mathbf{z})$ might be any function in a very broad class of possible functions (for example, the class of all continuous functions), no parametric function $g_m(\mathbf{z})$ would be capable of approximating arbitrarily well every possible function $h(\mathbf{z})$. What is achieved by the universal approximation theorem is to show that if the number of parameters is allowed to be increased, the best possible approximation of $h(\mathbf{z})$ will be progressively better, with the error converging to 0 in the limit for the number of parameters approaching ∞.

In the specific case of metaparametric neural networks, the class of parametric functions is a composition of two other classes of parametric functions as stated in Definition 1. Consequently, the number of input variables for the resulting function will be the sum of the number of input variables for the original functions. As shown in equation (3.5), given two classes of parametric functions that individually have universal approximation, it is possible to associate them so that the resulting class of parametric functions will not have universal approximation property in the resulting function space with increased input dimension. Therefore, it is necessary to prove an universal approximation theorem for MNNs in order to guarantee that the MNN structure does not impose any restrictions to the functions being modeled. The proof provided in Section 3.2.2 follows that general problem statement provided in the present section, but the input variable $\mathbf{z}$ is split into a $\mathbf{x}$ component and a $\mathbf{y}$ component so that each component represents the input variables of one of the parametric functions in the MNN.

The definition of $\mathcal{Z}$ is often restricted to sets with a certain property, like compact sets in Theorem 1. These restrictions simplify the demonstration, but they do not completely restrict the use of the result of the theorem results in sets that do not follow the restrictions. Indeed, the function of interest $h(\mathbf{z})$ might be defined over set $\mathcal{Z}' \subset \mathcal{Z}$, for example if one of the input variables is categorical. In this case, as long as it is possible to define an extension of the function $\tilde{h}(\mathbf{z}) : \mathcal{Z} \to \mathbb{R}^p$ so that $\tilde{h}(\mathbf{z}) = h(\mathbf{z})$ for any $\mathbf{z} \in \mathcal{Z}$, then $g_m(\mathbf{z}; \theta^{[m]})$ can universily approximate $\tilde{h}(\mathbf{z})$ and consequently it can also universally approximate $h(\mathbf{z})$.

3.2.2 Universal approximation proof

The main intuition behind Theorem 1 is that for any continuous function $h(\mathbf{x}, \mathbf{y})$ it is possible to define a set of parameters ψ so that $g(\mathbf{y}; \psi)$ is arbitrarily close to h. Then, it would suffice to use a neural network to represent the parameter ψ as a function of $\mathbf{x}$. However, there are important conditions that must be taken into account to assure the validity of this procedure. This can be summarized as follows: Let $g(\mathbf{y}; \psi)$ be a class of parametric functions that can approximate any continuous function provided that the number of parameters ψ is enough to achieve the required accuracy. An MNN would employ this function to approximate $h(\mathbf{x}, \mathbf{y})$ by making ψ the outputs of a neural network with inputs $\mathbf{x}$. Indeed, for any value of $\mathbf{x}$, there is a set of parameters ψ that will make $g(\mathbf{y}; \psi)$ satisfactorily close to $h(\mathbf{x}, \mathbf{y})$. In order to guarantee the MNN will approximate $h(\mathbf{x}, \mathbf{y})$ satisfactorily, three conditions must be met:

1. the number of parameters ψ must be the same for all values of $\mathbf{x}$. This condition is guaranteed by Lemma 2.

2. the function $\psi(\mathbf{x})$ (defined as the value of ψ that will make $g(\mathbf{y}, \psi)$ approximate $h(\mathbf{x}, \mathbf{y})$ for each value of $\mathbf{x}$) must be continuous so that it is possible to guarantee a neural network will approximate it arbitrarily well. This condition is guaranteed by Lemma 1.

3. the change in $g(\mathbf{y}; \psi)$ introduced by the neural network approximation of ψ must be small enough so that $g(\mathbf{y}; \psi(\mathbf{x}))$ is still a satisfactory approximation of $h(\mathbf{x}, \mathbf{y})$. This condition is guaranteed by Theorem 1.

We first provide the following definitions that will be used in all results in this section:

- $\mathcal{Z}$: a convex compact subset of $\mathbb{R}^{N_x+N_y}$;
- $\mathcal{X}$: the set of all $\mathbf{x} \in \mathbb{R}^{N_x}$ so that there is a point $(\mathbf{x}, \mathbf{y}) \in \mathcal{Z}$;
- $\mathcal{Y}$: the set of all $\mathbf{y} \in \mathbb{R}^{N_y}$ so that there is a point $(\mathbf{x}, \mathbf{y}) \in \mathcal{Z}$;
- $\mathcal{Y}(\mathbf{x})$: the set of all $\mathbf{y} \in \mathbb{R}^{N_y}$ so that $(\mathbf{x}, \mathbf{y}) \in \mathcal{Z}$;
- $\ell(\mathbf{x}; \psi)$: the supremum over $\mathbf{y}$ of the distance between $g(\mathbf{y}, \psi)$ and $h(\mathbf{x}, \mathbf{y})$,

as shown in equation (3.6):

$$\ell(\mathbf{x};\psi) = \sup_{\mathbf{y}\in\mathcal{Y}(\mathbf{x})} \|g(\mathbf{y},\psi) - h(\mathbf{x},\mathbf{y})\|, \tag{3.6}$$

where $\|\cdot\|$ is the $\ell^2 -$ norm.

Lemma 1. *Given a continuous function $h(\mathbf{x},\mathbf{y}) : \mathcal{Z} \to \mathbb{R}^{N_h}$ and a continuous parametric function $g(\mathbf{y};\psi) : \mathcal{Y} \times \mathbb{R}^n \to \mathbb{R}^{N_h}$, and provided that:*

(i) $g(\mathbf{y}, \alpha\psi_1 + (1-\alpha)\psi_2) = \alpha g(\mathbf{y},\psi_1) + (1-\alpha)g(\mathbf{y},\psi_2)$, for any $\psi_1, \psi_2 \in \mathbb{R}^n$, any $\alpha \in [0,1]$, and any $\mathbf{y} \in \mathcal{Y}$;

(ii) For any $\mathbf{x} \in \mathcal{X}$, there exists a function $\psi(\mathbf{x}) : \mathcal{X} \to \mathbb{R}^n$ so that $\ell(\mathbf{x};\psi(\mathbf{x})) < \eta$.

It is possible to build a continuous function $\psi_c(\mathbf{x}) : \mathcal{X} \to \mathbb{R}^n$ so that $\ell(\mathbf{x};\psi_c(\mathbf{x})) \leq \eta$ for any $\mathbf{x} \in \mathcal{X}$.

Proof. Let $\mathcal{P}_\eta(\mathbf{x})$ be the set of all $\psi \in \mathbb{R}^n$ for which $\ell(\mathbf{x},\psi) \leq \eta$. From the lemma condition (ii), $\mathcal{P}_\eta(\mathbf{x})$ is not empty. Since $\mathcal{Z}$ is compact and convex, the boundary of $\mathcal{Y}(\mathbf{x})$ is continuous in $\mathbf{x}$, thus $\ell(\mathbf{x};\psi)$ is continuous in $\mathbf{x}$ and ψ. Hence, the set $\mathcal{P}_\eta(\mathbf{x})$ is closed (i.e. contains all its limit points). Also, for any $\psi_1, \psi_2 \in \mathcal{P}_\eta(\mathbf{x})$, we have from the triangle inequality and from condition (i) that $\ell(\mathbf{x};\alpha\psi_1 + (1-\alpha)\psi_2) \leq \alpha\ell(\mathbf{x};\psi_1) + (1-\alpha)\ell(\mathbf{x};\psi_2) \leq \eta$, thus $\mathcal{P}_\eta(\mathbf{x})$ is convex.

In order to prove that there exists at least one continuous solution $\psi_c(\mathbf{x})$ to the inequality, we choose the particular solution $\psi_{min}(\mathbf{x})$ that minimizes $\|\psi\|$ for every $\mathbf{x}$. For any $\mathbf{x} \in \mathcal{X}$ we have from the Best Approximation Theorem [71] that there is one unique point $\psi_{min}(\mathbf{x}) \in \mathcal{P}_\eta(\mathbf{x})$ that minimizes $\|\psi\|$. Therefore, $\psi_{min}(\mathbf{x})$ is uniquely defined. We now show that $\psi_{min}(\mathbf{x})$ is continuous in $\mathcal{X}$. This has to be satisfied for each of the two scenarios for an arbitrary point $\mathbf{x}_0$ in $\mathcal{X}$:

- If $\ell(\mathbf{x}_0;\psi_{min}(\mathbf{x}_0)) < \eta$:

  From the continuity of $\ell(\mathbf{x}_0;\psi_{min}(\mathbf{x}_0))$, there exists a neighborhood of $\psi_{min}(\mathbf{x}_0)$ where all points ψ' satisfy $\ell(\mathbf{x}_0;\psi') < \eta$. Thus, in order to $\psi_{min}(\mathbf{x}_0)$ to have the minimum norm within this neighborhood, it must necessarily be the origin, $\mathbf{0}$.

  Also, from the continuity of $\ell(\mathbf{x};\mathbf{0})$, there exists a $\delta > 0$ so that $\ell(\mathbf{x};\mathbf{0}) < \eta$ for any $\mathbf{x}$ in the neighborhood of $\mathbf{x}_0$ defined by $\|\mathbf{x} - \mathbf{x}_0\| < \delta$. Thus in the

neighborhood of $\mathbf{x}_0$, $\psi_{min}(\mathbf{x}) = \mathbf{0}$, which implies that $\psi_{min}(\mathbf{x})$ is continuous.

- If $\ell(\mathbf{x}_0; \psi_{min}(\mathbf{x}_0)) = \eta$:

  From lemma condition (ii), there exists; $\psi^* \neq \psi_{min}(\mathbf{x}_0)$ satisfying $\ell(\mathbf{x}_0; \psi^*) < \eta$.

  Let $\psi_\alpha = (1-\alpha)\psi_{min}(\mathbf{x}_0) + \alpha\psi^*$. It follows from condition (i) that $g(\mathbf{y}, \psi_\alpha) = (1-\alpha)g(\mathbf{y}, \psi_{min}(\mathbf{x}_0)) + \alpha g(\mathbf{y}, \psi^*)$. Thus, from the triangle inequalities we have:

  1. $\ell(\mathbf{x}_0; \psi_\alpha) \leq (1-\alpha)\ell(\mathbf{x}_0; \psi_{min}(\mathbf{x}_0)) + \alpha\ell(\mathbf{x}_0; \psi^*) < \eta$ for any $\alpha \in (0,1)$;
  2. $\ell(\mathbf{x}_0; \psi_\alpha) \geq (1+|\alpha|)\ell(\mathbf{x}_0; \psi_{min}(\mathbf{x}_0)) - |\alpha|\ell(\mathbf{x}_0; \psi^*) > \eta$ for any $\alpha < 0$.

  Then, for any $\epsilon > 0$, there exists ψ^+ and ψ^- satisfying: $\|\psi^+ - \psi_{min}(\mathbf{x}_0)\| < \epsilon$; $\|\psi^- - \psi_{min}(\mathbf{x}_0)\| < \epsilon$; $\ell(\mathbf{x}; \psi^+) > \eta$; and $\ell(\mathbf{x}; \psi^-) < \eta$;

  From the continuity of $\ell(\mathbf{x}; \psi)$, there is $r_0(\epsilon) > 0$ so that $\ell(\mathbf{x}; \psi^+) > \eta$ and $\ell(\mathbf{x}; \psi^-) < \eta$ for any $\mathbf{x}$ with $\|\mathbf{x} - \mathbf{x}_0\| < r_0(\epsilon)$. Then, from the Intermediate Value Theorem, it is possible to define $\psi_\eta(\mathbf{x}; \epsilon)$ so that $\|\psi_\eta(\mathbf{x}; \epsilon) - \psi_{min}(\mathbf{x}_0)\| < \epsilon$ and $\ell(\mathbf{x}; \psi_\eta(\mathbf{x}; \epsilon)) = \eta$ for any $\mathbf{x}$ with $\|\mathbf{x} - \mathbf{x}_0\| < r_0(\epsilon)$. Thus, for any $\mathbf{x}$ with $\|\mathbf{x} - \mathbf{x}_0\| < r_0(\epsilon)$,

  $$\|\psi_{min}(\mathbf{x})\| < \|\psi_{min}(\mathbf{x}_0)\| + \epsilon\,. \tag{3.7}$$

  Assuming by contradiction that $\psi_{min}(\mathbf{x}_0)$ is not continuous in $\mathbf{x}_0$, there exists $K > 0$ so that for any $\delta > 0$ there exists a $\mathbf{x}'$ with $\|\psi_{min}(\mathbf{x}') - \psi_{min}(\mathbf{x}_0)\| > K$ and $\|\mathbf{x}' - \mathbf{x}_0\| < \delta$. Let $\delta_0 > 0$ and $\delta_j = \min(\delta_0/j, r_0(K/j))$, then there exists $\mathbf{x}'_j$ so that $\|\psi_{min}(\mathbf{x}'_j) - \psi_{min}(\mathbf{x}_0)\| > K$ and $\|\mathbf{x}' - \mathbf{x}_0\| < \delta_j$. Thus $\|\psi_{min}(\mathbf{x}'_j)\|$ is limited by equation (3.7) and from the Bolzano-Weierstrass theorem, there exists a convergent subsequence so that:

  $$\lim_{k\to\infty} \psi_{min}(\mathbf{x}'_{j_k}) = \psi^*\,. \tag{3.8}$$

  From the continuity of the norm, we have

  $$\lim_{k\to\infty} \|\psi_{min}(\mathbf{x}'_{j_k})\| = \|\psi^*\| \leq \|\psi_{min}(\mathbf{x}_0)\|\,, \tag{3.9}$$

but since $\ell(\psi_{min}(\mathbf{x}'_{j_k}), \mathbf{x}'_{j_k}) \leq \eta$ and $\ell(\mathbf{x}, \psi)$ is continuous:

$$\lim_{k\to\infty} \ell(\mathbf{x}'_{j_k}; \psi_{min}(\mathbf{x}'_{j_k})) = \ell(\mathbf{x}_0; \psi^*) \leq \eta\,. \tag{3.10}$$

Therefore, ψ^* is the minimum norm solution to $\ell(\mathbf{x}; \psi) \leq \eta$ and from equation (3.8) we have that $\lim_{k\to\infty} \psi_{min}(\mathbf{x}'_{j_k}) = \psi_{min}(\mathbf{x}_0)$. This is contradictory with the assumption that $\|\psi_{min}(\mathbf{x}'_j) - \psi_{min}(\mathbf{x}_0)\| > K$ for any $j > 0$. Thus, the assumption is false and $\psi_{min}(\mathbf{x})$ must be continuous in $\mathbf{x}_0$.

□

Lemma 2. *Given a continuous function $h(\mathbf{x}, \mathbf{y}) : \mathcal{Z} \to \mathbb{R}^{N_h}$, an infinite and strictly increasing sequence of positive integers, $(n_0, n_1, ...)$, and a class of continuous parametric functions in which for every n_j on that sequence there is a function with n_j parameters, $g(\mathbf{y}; \psi^{(n_j)}) : \mathcal{Y} \times \mathbb{R}^{n_j} \to \mathbb{R}^{N_h}$. And provided that:*

1. *$g(\mathbf{y}, \alpha\psi_1 + (1-\alpha)\psi_2) = \alpha g(\mathbf{y}, \psi_1) + (1-\alpha) g(\mathbf{y}, \psi_2)$ for any $\psi_1, \psi_2 \in \mathbb{R}^{n_j}$, any $\alpha \in [0, 1]$, and any $\mathbf{y} \in \mathcal{Y}$;*

2. *for any $\mathbf{x} \in \mathcal{X}$ and any $\epsilon > 0$, there exists a positive integer $N(\mathbf{x})$ so that for any $n_j \geq N(\mathbf{x})$ there exists $\hat{\psi}^{(n_j)}(\mathbf{x}) \in \mathbb{R}^{n_j}$ for which $\ell(\mathbf{x}; \hat{\psi}^{(n_j)}(\mathbf{x})) < \epsilon$.*

We have that for any $\epsilon > 0$, there is a positive integer N^* so that for any $n_j \geq N^*$, there exists a continuous function $\hat{\psi}_c^{(n_j)}(\mathbf{x}) : \mathcal{X} \to \mathbb{R}^{n_j}$ for which $\ell(\mathbf{x}; \hat{\psi}_c^{(n_j)}(\mathbf{x})) \leq \epsilon$.

Proof. Assuming by contradiction that $N(\mathbf{x})$ is not bounded, we can build an infinite sequence $\mathbf{x}_k$ so that $N(\mathbf{x}_k) > k$. Since $\mathcal{X}$ is compact, from the Bolzano-Weierstrass theorem there is a subsequence $\mathbf{x}_{k_j}$ that converges to some value $\mathbf{x}^* \in \mathcal{X}$. But for any $n_j \geq N(\mathbf{x}^*)$ there is $\hat{\psi}^{(n_j)}(\mathbf{x}^*) \in \mathbb{R}^{n_j}$ for which $\ell(\mathbf{x}^*, \hat{\psi}^{(n_j)}(\mathbf{x}^*)) < \epsilon$. From the continuity of $\ell(\mathbf{x}, \hat{\psi}^{(n)}(\mathbf{x}))$, we have that there is $\delta > 0$ so that $\ell(\mathbf{x}, \hat{\psi}^{(n)}(\mathbf{x})) < \epsilon$ for any $\mathbf{x}$ with $\|\mathbf{x} - \mathbf{x}^*\| < \delta$. This is contradictory with $N(\mathbf{x}_{k_j})$ diverging to ∞, thus $N(\mathbf{x})$ is bounded.

Now, let N^* be an upper bound of $N(\mathbf{x})$. From Lemma 1, it is possible to define for any $n_j \geq N$ a continuous function $\hat{\psi}_c^{(n_j)}(\mathbf{x})$ so that $\ell(\mathbf{x}; \hat{\psi}_c^{(n_j)}(\mathbf{x})) \leq \epsilon$ for any $\mathbf{x} \in \mathcal{X}$. □

Theorem 1 (Universal Approximation Theorem for Metaparametric Neural Networks)**.** *Given:*

- $(n_0, n_1, ...)$: *an infinite and strictly increasing sequence of positive integers for which a class of continuous parametric functions is defined,* $g(\mathbf{y}; \psi^{(n_j)}) : \mathcal{Y} \times \mathcal{P}^{(n_j)} \to \mathbb{R}^{N_h}$, *where* $\mathcal{P}^{(n_j)}$ *is a convex subset of* $\mathbb{R}^{n_j}$;

- $(m_0[n_j], m_1[n_j], ...)$: *an infinite and strictly increasing sequence of positive integers that might depend or not on* n_j, *so that a class of parametric functions is defined,* $\psi^{(n_j, m_k[n_j])}(\mathbf{y}; \theta^{(m_k[n_j])}) : \mathcal{X} \times \mathcal{T}^{(m_k[n_j])} \to \mathcal{P}^{(n_j)}$, *where* $\mathcal{T}^{(m_k[n_j], n_j)}$ *is a convex subset of* $\mathbb{R}^{m_k[n_j]}$.

And provided that:

1. $g(\mathbf{y}, \alpha\psi_1^{(n_j)} + (1-\alpha)\psi_2^{(n_j)}) = \alpha g(\mathbf{y}, \psi_1^{(n_j)}) + (1-\alpha) g(\mathbf{y}, \psi_2^{(n_j)})$ *for any* $\psi_1, \psi_2 \in \mathcal{P}^{(n_j)}$, *any* $\alpha \in [0,1]$, *and any* $\mathbf{y} \in \mathcal{Y}$;

2. *for any continuous function* $f(\mathbf{y}) : \mathcal{Y} \to \mathbb{R}^{N_h}$ *and any* $\epsilon > 0$, *there exists a positive integer* N *so that for any* $n_j \geq N$ *there exists* $\hat{\psi}^{(n_j)} \in \mathcal{P}^{(n_j)}$ *for which* $\sup_{\mathbf{y} \in \mathcal{Y}} \| g^{(n_j)}(\mathbf{y}, \hat{\psi}^{(n_j)}) - f(\mathbf{y}) \| < \epsilon$;

3. *for any* n_j, *any continuous function* $f(\mathbf{x}) : \mathcal{X} \to \mathbb{R}^{n_j}$ *and any* $\epsilon > 0$, *there exists a positive integer* $M[n_j]$ *so that for any* $m_k[n_j] \geq M[n_j]$ *there exists* $\hat{\theta}^{(m_k[n_j])} \in \mathcal{T}^{(m_k[n_j])}$ *for which* $\sup_{\mathbf{x} \in \mathcal{X}} \| \psi^{(n_j, m_k[n_j])}(\mathbf{x}, \hat{\theta}^{(m_k[n_j])}) - f(\mathbf{x}) \| < \epsilon$.

For any continuous function $h(\mathbf{x}, \mathbf{y}) : \mathcal{Z} \to \mathbb{R}^{N_h}$ *and any* $\epsilon > 0$, *there exist* J *and* $K[j]$ *so that for any* $j \geq J$ *and* $k \geq K[j]$ *there exists* $\hat{\theta} \in \mathcal{T}^{(m_k[n_j])}$ *so that:*

$$\sup_{(\mathbf{x}, \mathbf{y}) \in \mathcal{Z}} \| g^{(n_j)}(\mathbf{y}, \psi^{(n_j, m_k[n_j])}(\mathbf{x}, \hat{\theta}^{(m_k[n_j])})) - h(\mathbf{x}, \mathbf{y}) \| < \epsilon \,. \tag{3.11}$$

Proof. In this proof, we adopt the following notation:

$$\mathcal{L}\left(\psi^{(n_j)}(\cdot)\right) = \sup_{(\mathbf{x}, \mathbf{y}) \in \mathcal{Z}} \left\| g^{(n_j)}\left(\mathbf{y}, \psi^{(n_j)}(\mathbf{x})\right) - h(\mathbf{x}, \mathbf{y}) \right\| . \tag{3.12}$$

From Lemma 2, for any $\epsilon/2 > 0$ there is J so that for any $j \geq J$ there is a continuous function $\hat{\psi}^{(n_j)}(\mathbf{x}) : \mathcal{X} \to \mathcal{T}^{(n_j)}$ so that $\mathcal{L}(\mathbf{x}, \hat{\psi}^{(n_j)}(\cdot)) \leq \epsilon/2$.

Since $g^{(n_j)}(\cdot)$ is bounded and linear in $\psi^{(n_j)}(\cdot)$, $\mathcal{L}(\psi^{(n_j)}(\cdot))$ will be continuous in $\psi^{(n_j)}(\cdot)$. Thus, for any $\epsilon > 0$ and $j \geq J$ there is $\delta^{n_j}(\epsilon) > 0$ so that for any $\psi^{(n_j)}(\cdot)$ with $\sup_{\mathbf{x} \in \mathcal{X}} \| \psi^{(n_j)}(\mathbf{x}) - \hat{\psi}^{(n_j)}(\mathbf{x}) \| < \delta^{(n_j)}(\epsilon)$ we have $\mathcal{L}(\psi^{(n_j)}(\cdot)) < \epsilon$.

Finally, from the theorem conditions there is K so that for any $k \geq K$ there is

$\hat{\theta}^{(m_k[n_j])} \in \mathcal{T}^{(m_k[n_j])}$ for which $\sup_{\mathbf{x}\in\mathcal{X}} \|\psi^{(n_j,m_k[n_j])}(\mathbf{x}, \hat{\theta}^{(m_k[n_j])}) - \hat{\psi}^{(n_j)}(\mathbf{x})\| < \delta^{(n_j)}(\epsilon)$. □

3.2.3 Nested MNNs

From Theorem 1, it is known that an MNN can approximate any continuous function equally well as a neural network. An immediate consequence from it is that an MNN can be used to replace the black-box block in another MNN.

Corollary 1. *Given a neural network $\psi(\mathbf{x}, \theta)$ and K parametrizations $g_k(\mathbf{y}_k, \psi_k)$ following the Theorem 1 conditions, it is possible to build a mataparametric neural network in the form $g_K(\mathbf{y}_K; g_{K-1}(\mathbf{y}_{K-1}; ...g_1(\mathbf{y}_1; \psi(\mathbf{x}, \theta))...))$ that universally approximates any continuous function defined in the closed bounded subset $\mathcal{Z}$.*

This does not serve as a solution for completely eliminating the need of a neural network component in the MNN model because, if this procedure is performed iteratively until all input variables are captured by other parametric representations, this would make the number of parameters in the model grow exponentially. Nonetheless, this property is useful when there is some advantage in handling different input variables with different classes of parametric functions.

3.3 Possible choices for the blocks composing a metaparametric neural network

Now that Theorem 1 has been proven, we proceed to showing particular choices of parametrizations $g_\psi(\mathbf{y})$ that fulfils the conditions in the theorem so that the resulting MNN will represent arbitrarily well any continuous function in a compact subset of $\mathbb{R}^{N_x+N_y}$.

3.3.1 Basis function representations

We first analyze the case in which the parametrization $g(\cdot)$ is given by a set of basis function with arbitrary real-valued weights. In a basis function representation, a basis set with n functions of $\mathbf{y}$, $\nu_n(\mathbf{y})$, is used to achieve the parametric representation $g(\mathbf{y}; \psi)$ which is given by the linear combination of $\nu_n(\mathbf{y})$ with

weights ψ_n. This representation is expressed by equation:

$$g(\mathbf{y};\psi) = \sum_{n=1}^{N} \psi_n \nu_n(\mathbf{y}) \,. \tag{3.13}$$

When using basis function representations in the explicit block of an MNN, the first block of the MNN will have an arbitrary real valued output. There are numerous possible choices of basis function that can be used in the explicit layer of an MNN model and the best choice will depend on the application.

Polynomial basis functions

In polynomial basis functions representations, the basis $\nu_n(\mathbf{y})$ are defined as polynomial functions. The use of polynomials basis functions is supported by Taylor's Theorem [72], which provides a guarantee that any continuously differentiable function can be approximated arbitrarily well by an M-dimensional polynomial of order N in the form:

$$g(\mathbf{y};\psi) = \sum_{(n_1,\ldots,n_M)|n_1+\ldots+n_M \leq N} \left[\psi_{n_1,\ldots,n_M} \prod_{m=1}^{M} y_m n_m \right] . \tag{3.14}$$

The most notable property of the Taylor series relationship between the coefficients $\psi_{n_1,\ldots,n_m}$ with the partial derivatives of the function.

Numerous other polynomial basis functions have been proposed, most commonly with the purpose of achieving an orthogonal basis. Examples of orthogonal polynomial basis functions include Jacobi polynomials, Laguerre polynomials and Hermite polynomials.

Fourier series

The Fourier series basis function representation is given by:

$$g(y;\mathbf{a},\mathbf{b}) = a_0 \sum_{n=1}^{N} \left[a_n cos(2\pi n f y) + b_n sin(2\pi n f y) \right] , \tag{3.15}$$

or in its multidimensional version by:

$$g(\mathbf{y};\psi) = \sum_{n_1=-N}^{N} \dots \sum_{n_M=-N}^{N} \psi_{n_1,\dots,n_M} \exp\left(i \sum_{m=1}^{M} \omega_m n_m y_m \right) . \tag{3.16}$$

When using Fourier series representations in MNNs, it is important to notice that the standard formulation of Fourier series assumes that the target function is periodic. If an MNN would be used in an application where the target function was restricted to be periodic, this could be a good way of embedding this restriction into the model. However, in the present work and most likely in most other applications, this is not the case and this restriction can be avoided by forcing the function to be symmetric and only using half the interval where it is defined.

Spline basis functions

Splines are functions that are often used to interpolate discrete samples, which can be viewed as a low dimensional machine learning problem. They are defined as piecewise polynomial functions, ie. the estimation interval is split into sub-intervals inside which the splines are low order polynomials and in the sub-interval intersections there is a discontinuity in the highest non-zero component of the polynomial. These sub-intervals are defined by a set of points that corresponds to the intersection between them and are typically called knots. Spline basis functions are mainly characterized by the fact that the basis functions are concentrated in limited sub-intervals of the function domain, being equal to 0 outside this sub-interval. This property is extremely useful in machine learning applications since it is often the case that the correlation between points in a target function is large for neighbor points and tend to vanish for points that are far apart. This property is better understood in opposition to polynomial and Fourier series basis functions where each basis function has non-null components in all regions of the input domain; there if only one region of the function is to be changed this requires jointly changing all coefficients in a way that produces the desired change while minimizing the effect in other regions of the function domain.

Despite not being typically classified as a spline representation, the simplest set of basis functions that can be classified as splines is the piecewise-constant. There, each basis function is constant in a particular sub-interval and null outside it, thus being a 0-order polynomial. This results in a piecewise-constant function

with a stair-like shape with discontinuities in the knots.

The second simplest set of spline basis functions are the piecewise-linear. There, each basis function has a triangular shape, being non-null in two adjacent intervals and null in the remaining of the domain. The representation that results from this basis set is a piecewise-linear function that is linear within each sub-interval and has a discontinuity in the first derivative in the knots.

For any natural number, it is possible to define a different set of splines basis functions using polynomials of that order and making the discontinuity only happen in the highest order. For each order, it is possible to define a set of basis function that is spatially localized, helping estimation to be independent across different regions of the function domain. In practice, low order splines are the most useful since the higher the order of the splines the larger the number of sub-intervals where each basis function is non-null, which makes the behaviour of the basis functions approach that of polynomial basis functions as the spline order increases.

Another set of splines basis functions that commonly used is the third order polynomial, commonly referred to as natural cubic splines. They have previously been represented by a set of non-localized functions that have second and third order components within a few sub-intervals and are linear in the remaining of the function domain [22]. This representation is not ideal for neural networks since the magnitude of each function in a particular point might differ substantially, making the scale for each coefficient different, which is problematic for gradient based optimization. We propose an alternative representation that restricts the number of functions that is non-null within each sub-interval. Given knots $t_p, p \in \{0, \dots, P\}$, this basis is denoted by:

$$v_p(t) = \begin{cases} t + (t_0 + t_1 - 2t_2 - g_0(t))/3 & , p = 0 , \\ g_P(t)/3 & , p = 1 , \\ 1 - h_0(t) & , p = 2 , \\ h_{P-3}(t) & , p = 3 , \\ h_{p-3}(t) - h_{p-4}(t) & , p > 3 , \end{cases} \tag{3.17}$$

where $p \in \{0, \ldots, P\}$ and:

$$h_p(t) = \frac{g_p(t) - g_{p+1}(t)}{(t_{p+3} - t_p)}, \tag{3.18}$$

with $g_p(t)$ being the basis set used in [22]:

$$g_p(t) = -\frac{\max(0, t - t_{p+1})^3}{(t_{p+2} - t_{p+1})(t_{p+1} - t_p)} + \frac{\max(0, t - t_p)^3}{(t_{p+2} - t_p)(t_{p+1} - t_p)} \tag{3.19}$$

$$+ \frac{\max(0, t - t_{p+2})^3}{(t_{p+2} - t_p)(t_{p+2} - t_{p+1})}. \tag{3.20}$$

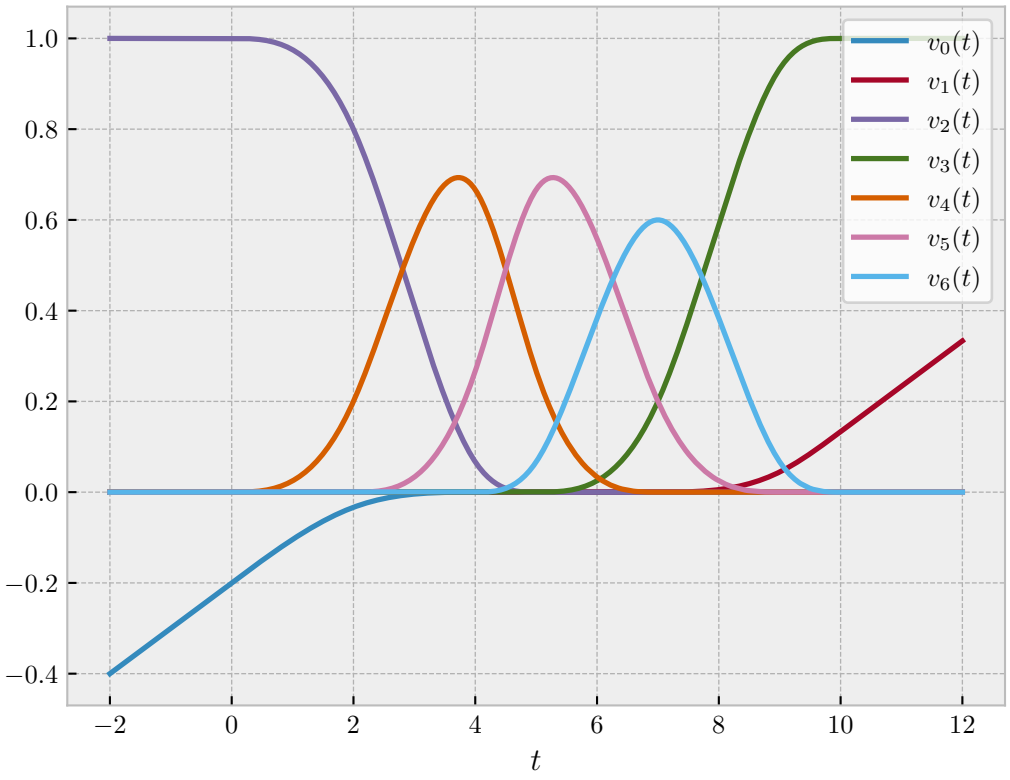


Figure 3.2.: Natural cubic splines basis set for knots defined at points 0, 2, 4, 5, 7, 9 and 10.

Figure 3.2 illustrates this basis for a particular set of knots. In particular, functions $v_0(t)$ and $v_{P-2}(t)$ are smooth functions that transition between 0 an 1 in the interval delimited by the four first or last knots respectively, and is constant elsewhere. Functions $v_p(t), p \in \{1, \ldots, P-3\}$ are smooth functions that are positive in the interval (t_{p-1}, t_{p+3}) and null elsewhere.

3.3.2 Mixture model representation

In addition to basis function representations, we also propose the use of mixture models as the explicit layer of MNN models. Mixture models are typically used in statistics in order to describe quantities in populations where for each individual, the target quantity follows a different probability distribution. Given that mixture models were designed to model compositions of probability distributions, the obey the restriction that a probability cannot be negative and must have an aggregate value of 1. As a result, the difference between a basis functions representations and mixture models is that in mixture models the weights are restricted so that all the weights are non-negative and their weight is equal to 1.

Despite being originally developed to model probability distributions, mixture models can be extremely useful for regression models when used within the MNN framework. There, n different functions are used in order to delimit possible functional relationships between the explicit variables and the outcome and for every combination of implicit input variables the outcome will be given by the weighted average of those components with the weights being given by the output of the neural network block of the MNN model, which will have a softmax activation function. Because of this structure, each component of the mixture model should represent a realistic relationship between the explicit variables and the outcomes, different from the basis function MNNs where each basis function does not need to individually represent a realistic instance of the target function. For this reason, we define each component with a parametric representation. More specifically, each component is defined with a basis function representation and the outputs of the neural network block of the MNN are not used as the weights in the basis function representation but as weights in the weighted average of different components. This structure is represented in Figure 3.3.

With this representation, the resulting function space is given by an (n-1)-dimensional concave polytope (ie. a hyperspace version of a polyhedron) with n vertices given by the functions in the basis set. On the other hand, in basis function representations, the function space is given by a (n-1)-dimension hyperplane that intersects all functions in the basis set. This means that the resultant function in a mixture model representation is constrained by its components. Consequently, it is easier to inspect the resulting model and as long as the behaviour of each of the n components in the mixture model is consistent, the

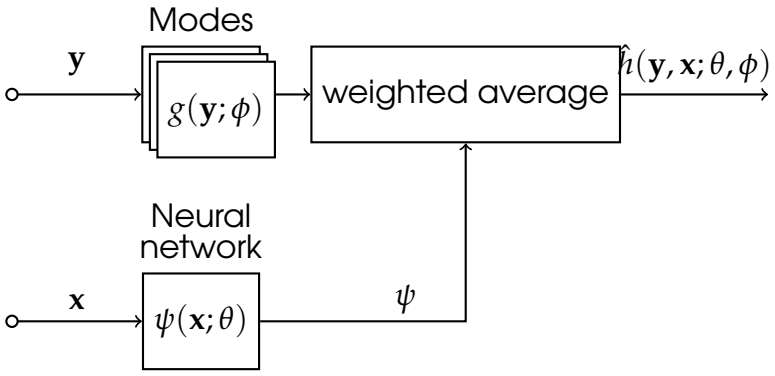


Figure 3.3.: Graphical description of a mixture model MNN.

resulting of the mixture model will be a combination of them and should also be consistent. Additionally, this can help to prevent overfitting since only a few components are available in a mixture model and it is unlikely that one of them will be used to overfit outliers.

Mixture models do not strictly fit the MNN structure used in the demonstration of Theorem 1. Nonetheless, it is possible to show the universal approximation property is maintained and we will now provide and ontline of how this could be demonstrated. Let each mode $g_n(\mathbf{y};\phi)$ be defined by a different combination of weights in a set of basis functions:

$$g_n(\mathbf{y};\phi) = \sum_{j=1}^{J} \phi_{n,j}\nu_j(\mathbf{y})\,. \tag{3.21}$$

The output of the MNN is then given by:

$$\hat{h}(\mathbf{y},\mathbf{x};\phi,\theta) = \sum_{n=1}^{N}\sum_{j=1}^{J} \psi_n(\mathbf{x};\theta)\phi_{n,j}\nu_j(\mathbf{y})\,, \tag{3.22}$$

where the activation function of the neural network is a softmax function, so that $\psi_n(\mathbf{x};\theta) > 0$ for any n and $\sum_{n=1}^{N}\psi_n(\mathbf{x};\theta) = 1$.

If the basis set $\nu_j(\mathbf{y})$ was used directly in an MNN model, the resulting neural network component of the model $\psi_j^*(\mathbf{x};\theta)$ would be a continuous function in a compact domain. Then, $\psi_j^*(\mathbf{x},\theta^*)$ will be superiorly and inferiorly bounded when approximating and continuous function $h(\mathbf{y},\mathbf{x})$. Denoting these bound as $\psi_{j,min}^*$

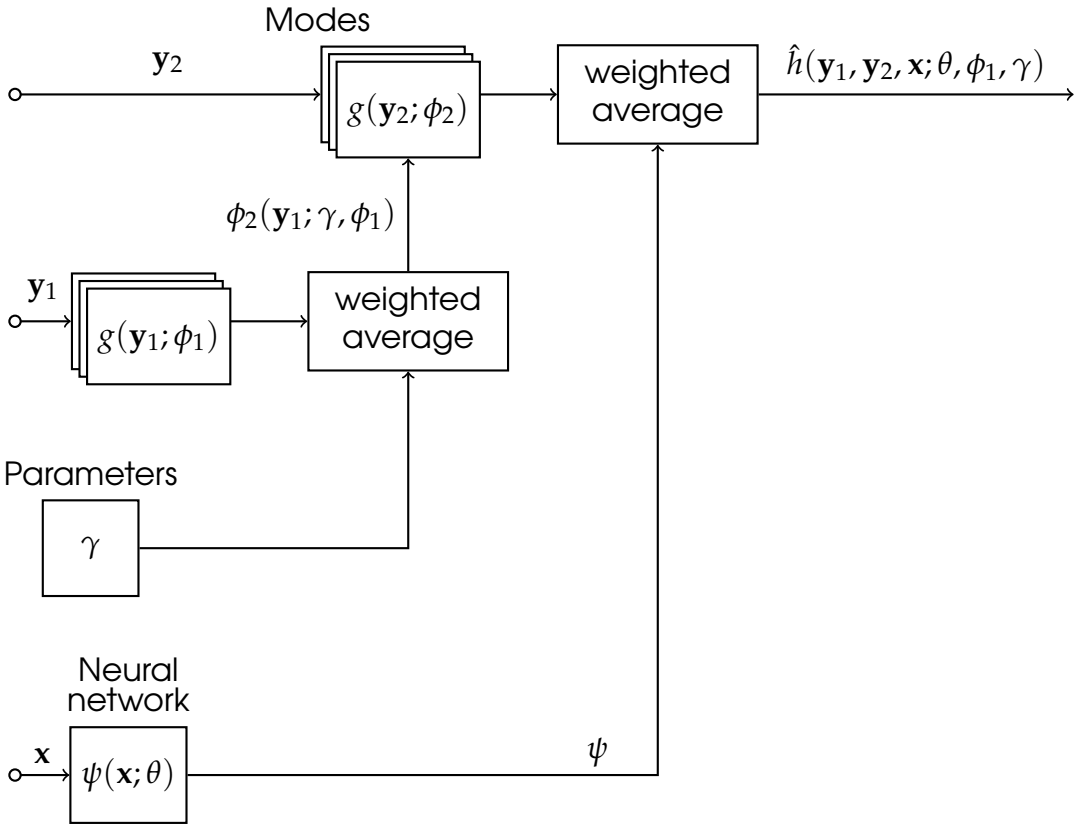


Figure 3.4.: Graphical description of a nested MNN.

and $\psi^*_{j,max}$, we define the following components for the mixture model:

$$g_n(\mathbf{y};\phi) = \sum_{j=1}^{J} \psi^*_{j,min}\nu_j(\mathbf{y}) + \begin{cases} 0 & ,n=1\,, \\ n \times (\psi^*_{n-1,max} - \psi^*_{n-1,min}) \times \nu_{n-1}(\mathbf{y}) & ,n>1\,. \end{cases} \tag{3.23}$$

Then, the mixture model will produce an accurate estimation of $h(\mathbf{y},\mathbf{x})$ if its coeffients are given by:

$$\psi_n(\mathbf{x};\theta) = \begin{cases} 1 - \sum_{k=2}^{J+1} \psi_k(\mathbf{x};\theta) & ,n=1\,, \\ \frac{\psi^*_{n-1}(\mathbf{x};\theta^*) - \psi^*_{n-1,min}}{\psi^*_{n-1,max} - \psi^*_{n-1,min}} & ,n>1\,. \end{cases} \tag{3.24}$$

From the construction of $\psi_n(\mathbf{x};\theta)$, it follows that it is continuous, and follow all constrains for the mixture model. Therefore, the neural network block of the mixture model MNN will be capable of approximating it with the required accuracy.

Nested MNNs

A further consequence of Theorem 1 is that in the mixture MNN formulation, it is possible to stack multiple mixture models as described in Figure 3.4. There, the basis function representation for $\mathbf{y}_1$ together with parameters γ work as a mixture MNN model where there is no implicit variables; therefore, the universal approximation property is valid for this block. Then, the output $\phi_2(\mathbf{y}_1; \gamma, \phi_1)$ is used as the implicit block for the basis function representation of $\mathbf{y}_2$ in the form of a basis function MNN, so the universal approximation property holds for the resulting block. Finally, the resulting block is used to replace the basis function representation in the standard mixture model MNN. Because of the universal approximation property, this block can approximate any continuous function of $\mathbf{y}_1$ and $\mathbf{y}_2$. Therefore, if this block is used to approximate the modes in a standard mixture model with both $\mathbf{y}_1$ and $\mathbf{y}_2$ as explicit variables, it will be possible to approximate any continuous function $h(\mathbf{y}_1, \mathbf{y}_2, \mathbf{x})$ and the universal approximation property is true for the entire nested model.

The advantage of this structure over the standard mixture model is that a bottleneck is introduced, so that the number of parameters in the explicit block will not grow exponentially with the number of explicit variables. One important feature of this bottleneck is that despite limiting the amount of information represented by the model, it does not interfere with the type of structure that can be represented by it. This bottleneck is analogous to a reduction of the number of hidden units in a neural network, where the number of parameters is reduced but there is no fundamental change in the structure of the model. In principle, this nested structure could be used to achieve a completely explicit MNN model without the need of representing any variable implicitly through a neural network block, but this possibility was not explored in detail in the present project and may be a theme for future works.

3.4 Relationship with other neural network architectures

The proposed architecture for the MNN resembles other existing structures in the literature and its relationship to them will be detailed in the present section. The first is the mixture of experts [73, 74]. There, assuming that a function $h(\mathbf{z}) : \mathcal{Z} \to \mathbb{R}^J$ is being modeled, multiple experts are used to model it in parallel, $\psi_1(\mathbf{z}), \ldots, \psi_M(\mathbf{z})$. Then, an additional neural network called gating function

$\rho(\mathbf{z})$ is used to weight each expert $\psi_m(\mathbf{z})$ with the goal of attributing a higher weight to the model that has the best performance in the specific point $\mathbf{z}$. In the training process, all experts and the gating function are trained at the same time forming a large joint model that outputs the average of the outcome of each expert weighted by the gating function. Similarly to the mixture of exterts, an MNN also has multiple estimations that are weighted using a function of input covariates. Indeed, if the choice of basis funcitons in an MNN is so that all bassis functions are non-netative and their sum is always 1, then they could be seen as a gating function. Despite this structural similarity, there is a fundamental difference which is that in the mixture of experts architecture both the experts and the gating function have the same input variables, whereas in the MNN architecture there is a separation of the inputs between implicit variables $\boldsymbol{x}$ and explicit variables $\boldsymbol{y}$. This separation is precisely what allows MNNs to fulfill their main purposes, which are to improve interpretability of the resultant model and to allow analytic operations like integration and derivative to be performed over the explicit variables. Indeed, this separation allows the use of basis functions given the reduced dimensionality of $\boldsymbol{y}$. This means that for any fixed value that is chosen for $\boldsymbol{x}$, the MNN will provide a basis function representation of its estimate of $h(\boldsymbol{x}, \boldsymbol{y})$. This representation can be interpreted easier than neural networks and also allows analytic derivative and integration.

Another class of models that resembles the MNN structure is the mixture density network [75], where a Gaussian mixture model of the outcome variable is obtained with its parameters being the outputs of a neural network with a different set of input variables:

$$f(y|\mathbf{x}) = \sum_{m=1}^{M} \rho_m(\mathbf{x}) \phi_{\mu_m(\mathbf{x}),\sigma_m(\mathbf{x})^2}(y)\,, \tag{3.25}$$

where for every m, $\phi_{\mu_m(\mathbf{x}),\sigma_m(\mathbf{x})^2}(y)$ if the Gaussian probability density function with mean $\mu_m(\mathbf{x})$ and variance $\sigma_m(\mathbf{x})^2$ and $\rho_m(\mathbf{x})$ is the weight for distribution m in the Gaussian mixture model. Although this architecture can be represented in terms of the MNN framework, it is a specific case of an MNN and not an equivalent to it. Indeed, the MNN framework allows the representation not only of a probability distribution over the outcome variable but also the representation of a generic function as the output of a neural network. Conversely, in mixture density networks, the arguments y of the Gaussian mixture is seen as the outcome of the model and not an input variable, with the aim of creating a probabilistic

representation of the outcome. The application of MNNs to survival modeling performed in the next chapters of this thesis provides practical results that illustrate well the improvements achieved with the MNN framework. First, in the PH-MNN model in Section 4.3.1, where the MNN is used to represent the hazard ratio, which is not a probability distribution over time, but a non-negative function of time that modulates the baseline hazard function. There, it would be possible to achieve an structure analogous to the DH-MNN model using a mixture density network, since there the outcome of the model is precisely a probability distribution. However, the PH-MNN model uses the separation between hazard ratio and baseline hazard function proposed in [2] which allows high frequency components of the time-to-event probability distribution to be modeled without covariate dependency. This has led to improved estimation accuracy as shown in Section 4.4. Second, in the nested PN-MNN model where the outcome is modeled with a hierarchical structure that improves interpretability as shown in Section 4.5. Third, in the transfer learning strategy proposed in Section 6.2. There, the separation of modes present in the nested PH-MNN model was used to reduce overfitting in a scenario where part of the input variables was only available in a reduced subset of the data.

4

Metaparametric neural networks for survival analysis

In the present chapter we show how the MNN framework can be applied to survival analysis. Additionally, we show that the main works done so far on the application of neural networks to survival analysis can be cast as restricted instances of the MNN survival models. The content of this chapter is derived from [70].

In current neural network models for survival analysis, it is possible to see important limitations caused by the black-box structure. Current extensions of statistical approaches to survival modeling impose limitations on the functional relationships to be modeled in order to allow better interpretability of the influence that time has on the event probability. Indeed, most neural network extensions of the proportional hazards model assume that the hazard ratio is time invariant, with the only exception being Cox-time [16] which is extremely computational costly. The extensions of the accelerated failure time model assume that the event probability conforms to particular families of probability distributions. Conversely, discrete time models compute the output for several numbers of time points in order to represent the functional relationship between the time and the event probability. This does not directly solve the lack of interpretability of the influence time has on the event probability, but divides the problems

into several smaller problems so that in each smaller problem this influence can be neglected. However, this approach restricts estimation to points included in the sample and increasing the sample size would lead to an increased number of parameters and possibly also the need for more data to train the model. Although it is possible to use interpolation methods to achieve estimations in intermediary points, it is preferable to employ a model that uses the same parametric representation during training so that the neural network is optimized to represent outcomes at the entire range of follow up times.

The MNN framework the will be applied to survival analysis in the present chapter allowing the generalization of several survival models. Indeed, it is shown in Section 4.1 that various survival models can be interpreted as particular instances of MNNs and this knowledge is used in Section 4.2 to extend existing models. This results in tree different models, each coming from a different class of models. The advantage of the MNN approach in the present work is shown experimentally in Section 4.4.2. Additionally, the introduction of a more interpretable structure to the model allows for further improvements, including the possibility of models that can be more easily scrutinized as will be shown in Section 4.5.2 and the possibility of combining datasets with different sets of input variables as will be shown in Section 6.2.

4.1 Interpretation of other neural network survival models in the form of MNNs

The MNN structure can be used as a generic framework for survival analysis and most survival models can be described as specific cases of it. The existing neural network extensions of the proportional hazards model can be cast in the metaparametric form. This is achieved by making for each event type j: $\lambda_j(t;\mathbf{x}) = g_j(t;\psi_j(\mathbf{x};\theta)) = \lambda_{0,j}(t)\exp(\psi_j(\mathbf{x};\theta))$, where $\lambda_{0,j}(t)$ is the baseline hazard function for event type j and $\psi_j(\mathbf{x};\theta)$ is the output of a neural network. In the case of Cox-Time [16], the follow-up time t is included as part of vector $\mathbf{x}$. Also, the original Cox proportional hazards model and both Fine & Gray [4] and Kalbfleish & Prentice [3] competing risks extensions of it can also be seen as a particular instance of it by restricting $\psi_j(\mathbf{x};\theta)$ to a linear model.

Similarly, the neural network versions of the AFT model [27] can also be ex-

pressed in a metaparametric form. This is done by making $g(t;\psi(\mathbf{x};\theta))$ a log-normal probability distribution with parameters $\mu = \psi_1(\mathbf{x};\theta)$ and $\log\sigma = \psi_2(\mathbf{x};\theta)$, where $\psi_{[1,2]}(\mathbf{x};\theta)$ are the outputs of a neural network. The original AFT model can also be seen as a particular instance of it by restricting $\psi_{[1,2]}(\mathbf{x};\theta)$ to a linear model.

The discrete time-interval models can also fit in a metaparametric structure, with the use of a series of Kronecker delta functions. This results in a cause-specific hazard function that is defined over a time interval, as follows: $\lambda_j[\kappa;\mathbf{x}] = g_j[\kappa;\psi_{j,[0,\ldots,K]}(\mathbf{x};\theta)] = \psi_{j,\kappa}(\mathbf{x};\theta)$, where κ is the index of a time interval and $\psi_{j,[0,\ldots,K]}(\mathbf{x};\theta)$ are the outputs of a neural network.

More importantly, the MNN framework can be used to formulate more generic models. This requires:

1. showing how the output of the MNN will describe the survival probability distribution, which is covered in Section 4.2;
2. making a choice of parametric function $g_j(t;\psi(\mathbf{x};\theta))$, which is covered in Section 4.2.5
3. estimating the parameters of the neural networks, which is covered in Section 4.3.

4.2 MNN survival modeling framework

As shown in Section 4.1, the metaparametric structure provides a formal generic framework for most neural network based survival models. Here, we exploit this finding to derive novel extensions for all three classes of survival models.

4.2.1 Proportional hazards metaparametric neural network (PH-MNN)

We define the PH-MNN with the expression:

$$\lambda_j(t,\mathbf{x}) = \lambda_{0,j}(t)\omega_j(t,\mathbf{x})\,, \tag{4.1}$$

where $\lambda_{0,j}(t)$ is the cause-specific baseline hazard function and $\omega_j(t, \mathbf{x})$ is the time-dependent hazard ratio, given by:

$$\omega_j(t, \mathbf{x}) = a\left(\sum_{k=1}^{K} \psi_{k,j}(\mathbf{x})\nu_k(t)\right), \tag{4.2}$$

where $\nu_k(t)$ is a set of basis functions over time; $\psi_{k,j}(\mathbf{x})$ are outputs of a neural network; and $a(\cdot)$ is a strictly positive function. Typically, $a(\cdot)$ is chosen to be the exponential function. The choice of the basis $\nu_k(t)$ and the function $a(\cdot)$ will strongly influence the model estimation procedure and its computational requirements. If the basis $\nu_k(t)$ is localized in time, being positive inside a finite interval and null outside it, the following simplified structure is useful:

$$\omega_j(t, \mathbf{x}) = \sum_{k=1}^{K} a\left(\psi_{k,j}(\mathbf{x})\right)\nu_k(t). \tag{4.3}$$

Here, the time localization and non-negativity of the basis is required to guarantee that $\omega_j(t, \mathbf{x}) \geq 0$. The variability in the amount of data for each type of event may dictate that we choose a different basis set $\nu_k(t)$ for each event type j.

The PH-MNN structure allows universal approximation of any continuous survival function given that the baseline hazard function $\lambda_{0,j}(t)$ is non-parametric, being able to approximate any continuous unidimentional function, while $\omega_j(t, \mathbf{x})$ is represented by an MNN, following the universal approximation property as shown in Theorem 1. Although the MNN block in the PH-MNN model has the restriction of being non-negative, which is achieved through the activation function $a(\cdot)$, this does not harm its universal approximation property since the hazard ratio to be estimated is required to be non-negative by definition.

4.2.2 Quantile regression metaparametric neural network (QR-MNN)

We define the QR-MNN quantile function as:

$$Q(\tau, \mathbf{x}) = \int_{u=0}^{-\log\tau} a\left(\sum_{k=1}^{K} \psi_k(\mathbf{x})\nu_k(u)\right) \mathrm{d}u, \tag{4.4}$$

where $Q(\tau, \mathbf{x}) = \inf\{t : 1 - S(t|\mathbf{x}) \geq \tau\}$. The metaparametric formulation must respect the constraint that $Q(\tau, \mathbf{x})$ should be strictly increasing with time. A suitable basis set $\nu_k(t)$ should provide an analytical expression for the integral in equa-

tion (4.4). Analogous to the PH-MNN model, an alternative parametrization can be obtained by placing the function $a(\cdot)$ inside the summation resulting in the following form:

$$Q(\tau, \mathbf{x}) = \sum_{k=1}^{K} a\left(\psi_k(\mathbf{x})\right) \int_{u=0}^{-\log \tau} \nu_k(u) \mathrm{d}u \,. \tag{4.5}$$

This makes analytical integration more simple. A competing risks extension can be achieved using a cause-specific quantile, which we define as $Q_j(\tau, \mathbf{x}) = \inf\{t : 1 - \exp(-\Lambda_j(t; \mathbf{x})) \geq \tau\}$, where $\Lambda_j(t; \mathbf{x}) = Pr[T_{event} < t; j|\mathbf{x}]$ is the cause-specific cumulative hazard function.

Note that either the quantile function or its competing risks extension fully specifies the event probability distribution and the correspondent hazard function can be retrieved from it:

$$\lambda_j(t, \mathbf{x}) = -\frac{\mathrm{d}}{\mathrm{d}t} \log\left[1 - Q_j^{-1}(t, \mathbf{x})\right] \,. \tag{4.6}$$

The QR-MNN model satisfies the universal approximation property since the derivative of its quantile function is represented by an MNN model. Here, the restriction for this derivative to be non-negative also does not hurt the universal approximation property since the derivative that is being estimated is required to be non-negative by definition.

4.2.3 Direct hazard metaparametric neural network (DH-MNN)

We define the DH-MNN as a continuous time extension of the discrete time-interval models. This is achieved with the following formulation:

$$\lambda_j(t, \mathbf{x}) = a\left(\sum_{k=1}^{K} \psi_{k,j}(\mathbf{x}) \nu_k(t)\right) \,, \tag{4.7}$$

where the function $a(\cdot)$ should be positive for the model to be coherent, in the sense that the hazard function is never negative. This is a direct functional representation of the hazard function and; therefore, can be termed as a direct hazard model.

An alternative formulation, as in the PH-MNN and QR-MNN, is:

$$\lambda_j(t, \mathbf{x}) = \sum_{k=1}^{K} a\left(\psi_{k,j}(\mathbf{x})\right) \nu_k(t)\,, \tag{4.8}$$

where the basis set $\nu_k(t)$ should be positive and localized in time.

In the DH-MNN, the hazard function, which is non-negative by definition is directly modeled by an MNN with activation function $a(\cdot)$ that guarantees non-negativity. This makes the universal approximation also valid for this model. Note that the DeepHit model [45], being a particular instance of the DH-MNN with Dirac delta activation function will not universally approximate the hazard ratio but will universally approximate the cumulative hazard function. Indeed, the DeepHit model was an important step towards the development of the MNN framework. Nonetheless, its use of a discrete basis function requires more unites than higher order polynomials to achieve a precise representation of the time dependency of the survival function, making it preferable to use other versions of the DH-MNN model.

4.2.4 General remarks

In all of the above models, an infinite set of basis functions can represent any square integrable function of time in a finite interval [76]. Restricting the number of basis functions to be finite has an effect analogous to eliminating the high frequency components of the target function. In practice, a sufficient approximation accuracy to any function can be achieved by a suitable finite set of basis functions. This approach provides a continuous and smooth representation of the target function, whilst reducing the required number of basis functions, and consequently the risk of overfitting. The extension of the proportional hazards model has the additional advantage of allowing the inclusion of high frequency components of the hazard function that are common to all values of $\mathbf{x}$, via the baseline hazard function.

4.2.5 Choice of basis functions

After presenting the application of the MNN framework to survival analysis, we now proceed to further describing the choices of basis function presented in

3.3.1 but now in the context of survival models. More specifically, we show how this choice affects the possibility of analytical computation in equations (4.2) to (4.8).

Piecewise constant basis functions

Given a set of time knots $[\bar{T}_0, \bar{T}_1, ..., \bar{T}_K]$, the set of basis functions for a piecewise constant model will be:

$$\nu_k(t) = \begin{cases} 1, \bar{T}_{k-1} \leq t < \bar{T}_k \\ 0, \text{otherwise}\,. \end{cases} \tag{4.9}$$

This set of basis functions is completely separated in time, simplifying model computation.

For a PH-MNN model, this basis choice makes equations (4.2) and (4.3) equivalent and removes the need to compute $a(\cdot)$ for each time point separately in the objective function. Also, this choice of basis functions allows analytical conversion between different representations of the event probability distribution for all MNN models, thereby reducing the computational cost of the estimation.

Although the computation is simpler than with other choices of basis functions, a smooth transition between intervals cannot be achieved, with discontinuities in the modeled hazard function despite the target function being smooth.

Piecewise linear basis functions

Given a set of time knots $[\bar{T}_0, \bar{T}_1, ..., \bar{T}_K]$, the set of basis functions for a piecewise constant model will be:

$$\nu_k(t) = \begin{cases} (t - \bar{T}_{k-1})/(\bar{T}_k - \bar{T}_{k-1}), \bar{T}_{k-1} \leq t < \bar{T}_k \\ (\bar{T}_{k+1} - t)/(\bar{T}_{k+1} - \bar{T}_k), \bar{T}_k \leq t < \bar{T}_{k+1}\,. \\ 0, \text{otherwise} \end{cases} \tag{4.10}$$

In contrast to the piecewise constant models, the basis functions are continuous. For the PH-MNN model, equations (4.2) and (4.3) are no longer equivalent. Although both formulations are possible, (4.3) will have smaller computational cost for estimation, as discussed in Section 4.3. The same is true of QR-MNN or

DH-MNN models and computation will be simplified with the use of equations (4.5) and (4.8) respectively. This choice of basis functions also allow analytic conversion among different representations of the event probability distribution, analogous to the piecewise constant basis functions.

Natural cubic splines

When used in the formulation provided in Section 3.3.1, it is also possible to use the model formulations in equations (4.3), (4.5) and (4.8).

Other basis functions

Other choices of basis functions are possible that make the resultant model smoother than in the piecewise models. These include higher order polynomials and Fourier basis functions. In this case, it is not possible to use the model formulations provided in equations (4.3), (4.5) and (4.8). Instead, a similar effect is achieved by making $a(y) = y^2$ in equations (4.2), (4.4) and (4.7). Given a set of unconstrained coefficients of y, a convolution property can be used to compute a set of coefficients that will produce y^2 in the same basis either in Fourier or in polynomial representations. If the convolution property is used, $\Lambda_j(t, \mathbf{x})$ can be computed analytically through the integration of each basis function individually. For a QR-MNN model, the inverse of the quantile function cannot be computed analytically, requiring a numerical approximation to be used.

4.2.6 Nested MNN for survival analysis

The nested MNN formulation proposed in Section 3.3.2 can be applied to the survival models proposed in Section 4.2. Here, the advantage of this formulation would be to allow easier scrutiny of all possible outcomes of the survival model as they unfold in time. In the present chapter we evaluate the nested version of the PH-MNN model (nested PH-MNN). This is done in Section 4.4 where its estimation error is compared to other models and in Section 4.5 where the practical impacts of the nested PH-MNN structure is further explored.

4.3 Estimation of MNN models

4.3.1 Proportional hazards metaparametric neural networks

The original estimation method for the proportional hazards model is the partial likelihood maximization [2], with competing risks extensions proposed by [3, 4, 77]. These estimators are compatible with time-dependent hazard ratios and require no further development for implementation with the PH-MNN structure, regardless of the different forms of the partial likelihood objective function. Nonetheless, existing proofs of asymptotic properties of the estimation cease to be valid in the case of the PH-MNN model. The theoretical background for the use of partial likelihood estimation in the PH-MNN model will be provided in Chapter 5.

In addition the the asymptotic properties of the estimator, special care is required in the implementation to avoid impractical computational cost and this is the focus of the present section. We show here the estimation procedure for the Cox partial likelihood estimator, but same procedure can be applied to other objective functions presented in Chapter 5.

The Cox partial log-likelihood is given by:

$$\mathcal{L} = \sum_n \mathcal{L}_n \,, \tag{4.11}$$

where:

$$\mathcal{L}_n = \left[\log \omega(t, \mathbf{x}_n) - \log \sum_{m=n}^{N} \omega(t, \mathbf{x}_m) \right] E_n \,. \tag{4.12}$$

There E_n indicates if an event has occurred to subject n at time T_n. For N subjects, the computational complexity of a training step is $\mathcal{O}(N^2 N_K + N(C_F + C_B))$, where N_K is the number of basis functions, and C_F and C_B are respectively the computational costs of feed-forward and back-propagation in the chosen neural network architecture. This is impractical for large datasets, and is avoided in traditional neural networks by mini-batch approximation or by on-line training [78]. Here, an extension of this technique is required since standard mini-batch approximation would still lead to a computational cost that grows linearly with N. This is achieved by training the data with two independent sets of mini batches:

1. the first containing an arbitrary set of subjects with size N_b;

2. and the second containing only uncensored subjects with size $\tilde{N}_b$.

The mini batch approximation of $\log \sum_{m=n}^{N} \omega(t, \mathbf{x}_m)$ is achieved by replacing the summation with the average of $\omega(t, \mathbf{x}_m)$ for all $\mathbf{x}_m$ in the mini batch 1. The approximation of $\mathcal{L}$ is given as the average of all $\mathcal{L}_n$ in mini batch 2. For simplicity, we normalize the log-likelihood by the number of uncensored subjects. Note that for each subject in mini batch 2, it is necessary to make an independent estimation within mini batch 1. Then, the cost of one training iteration becomes $\mathcal{O}(N_K N_b \tilde{N}_b + (N_b + \tilde{N}_b)(C_F + C_B))$.

The estimation of the baseline hazard function requires consideration of time variation. The Kalbfleish & Prentice estimator [11] and the Breslow estimator [10] both provide an analytical expression for the baseline hazard function, but assume a time-invariant proportionality factor and a single risk. However, Kalbfleish & Prentice can be extended by computing the cumulative hazard in the form:

$$\Lambda_j(\mathbf{x}, t) = \sum_{T_n<t; E_n=1; j_n=j} -\frac{\omega_j(\mathbf{x}, T_n)}{\omega_j(\mathbf{x}_n, T_n)} \log\left[1 - \frac{\omega_j(\mathbf{x}_n, T_n)}{\sum_{T_m \geq T_n} \omega_j(\mathbf{x}_m, T_n)}\right]. \tag{4.13}$$

Although the proof of maximum likelihood for this estimator provided in [11] does not extend directly to the case of the PH-MNN model, an extension of this proof is provided in Chapter 5. Note that if the model is based in equation (4.2), the computational cost of estimating the survival probability for one single subject after the model has been trained grows linearly with the training dataset size. If the model is based on equation (4.3), the summation in equation (4.13) can be rearranged as follows:

$$\begin{aligned}\Lambda_{0,j}(t) &= \sum_{T_n<t; E_n=1; j_n=j} -\frac{\sum_{k=1}^{K} a\left(\psi_{k,j}(\mathbf{x})\right) \nu_k(T_n)}{\omega_j(\mathbf{x}_n, T_n)} \log\left[1 - \frac{\omega_j(\mathbf{x}_n, T_n)}{\sum_{T_m \geq T_n} \omega_j(\mathbf{x}_m, T_n)}\right] \\ &= \sum_{k=1}^{K} a\left(\psi_{k,j}(\mathbf{x})\right) H_{k,j}(t), \end{aligned} \tag{4.14}$$

where:

$$H_{k,j}(t) = \sum_{T_n<t; E_n=1; j_n=j} -\frac{\nu_k(T_n)}{\omega_j(\mathbf{x}_n, T_n)} \log\left[1 - \frac{\omega_j(\mathbf{x}_n, T_n)}{\sum_{T_m \geq T_n} \omega_j(\mathbf{x}_m, T_n)}\right]. \tag{4.15}$$

There, $H_{k,j}(t)$ only needs to be computed once with the values for each T_n in the dataset being recorded while iterating through the summation. This makes com-

putational complexity of calculating the survival function for each new subject $\mathcal{O}(C_F + \log N)$.

4.3.2 Quantile regression metaparametric neural networks

Estimation in the QR-MNN model is performed by maximizing its log-likelihood, given by:

$$\mathcal{L} = \sum_{n=1}^{N} \left[E_n \log(\lambda_{j_n}(\mathbf{x}_n, T_n)) - \sum_{j=1}^{J} \Lambda_j(\mathbf{x}_n, T_n) \right], \tag{4.16}$$

where $\Lambda_j(\mathbf{x}, t) = \int_0^t \lambda_j(\mathbf{x}, \nu) \mathrm{d}\nu$ and the cause-specific hazard function $\lambda_j(\mathrm{x}, t)$ can be retrieved from the cause-specific quantile function in equation (4.6). Here, standard mini-batch approximation can be performed. Note that the estimation of this likelihood requires the computation of the inverse of the quantile function, so estimation will be impacted by the choice of basis functions as highlighted in Section 4.2.5. If the basis function is chosen to be piecewise constant or piecewise linear, the inverse of the quantile function can be computed analytically and the computational complexity of training a single batch will be $\mathcal{O}(N_b(C_F + C_B))$, where N_b is the size of the mini-batch, and C_F and C_B are respectively the costs of feed-forward and back-propagation in the chosen neural network architecture.

4.3.3 Direct hazard metaparametric neural networks

We estimate the DH-MNN model by maximizing its log-likelihood, given by equation (4.16) in Section 4.3.2. The computation of the likelihood is simplified if the version of the model in equation (4.8) is used, since the integral can be computed analytically. Here, the standard mini-batch approximation can also be performed. If the basis functions are chosen to be piecewise constant or piecewise linear, as detained in Section 4.2.5, the computational complexity of training a single batch will be $\mathcal{O}(N_b(C_F + C_B))$, as with the QR-MNN model.

4.4 Experimental comparison of MNNs with other models

In the present section we perform a comparison of baseline implementations of each type of MNN based survival model with other survival models.

4.4.1 Application to synthetic data modeling

We first provide an example of the application of the proposed models to estimate the cause-specific survival probability distribution in a synthetic dataset. The synthetic data used has two input covariate and two possible events, with the cause-specific hazard function being:

$$\lambda_1(t,\mathbf{x}) = 0.03(1+0.5\cos(2\pi t/10))\exp(\tan^{-1}(2x[0])\mathbb{1}(t<5)+\tan^{-1}(2x[1])\mathbb{1}(t>5))\,, \tag{4.17}$$

and

$$\lambda_2(t,\mathbf{x}) = 0.03(1+0.5\sin(2\pi t/10))\exp(\sin(x[1])\mathbb{1}(t<5)+\sin(x[0])\mathbb{1}(t>5))\,, \tag{4.18}$$

where x[0] and x[1] have independent normal distributions with mean equal to 0 and standard deviation equal to 1 and $\mathbb{1}(\cdot)$ is the indicator function, which takes the value of 1 when the argument is true and 0 otherwise.

In all neural network models, the same structure was used to compute $\psi_{k,j}(\boldsymbol{x})$, which included Gaussian dropout [79]. This structure is described in Fig. 4.1, reproduced with permission from IEEE.

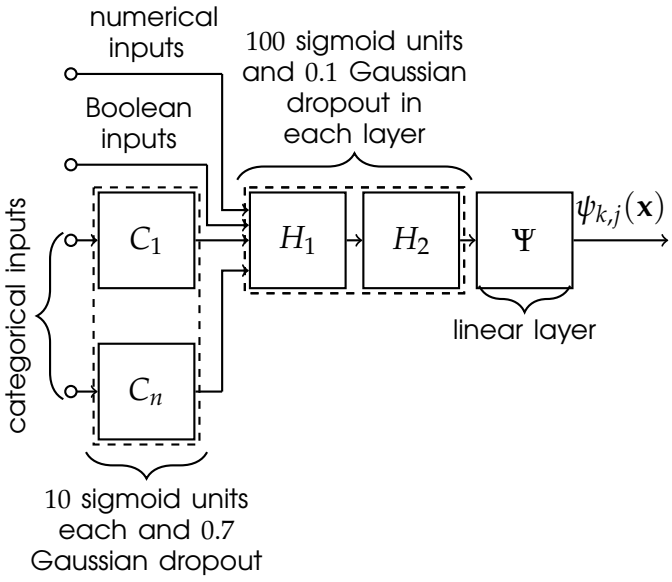


Figure 4.1.: Graphical description of the neural network structure applied in all models. [70] © 2021 IEEE.

The following models were compared:

- **PH-MNN**: with piecewise linear basis functions and time knots equally dis-

tributed in intervals of 2. The pseudo-code for this model is provided in Algorithm 5 in Appendix A.

- **QR-MNN**: with piecewise linear basis functions and quantile knots given by $\exp(-\Lambda_k)$ with $\Lambda_k \in \{0.01, 0.03, 0.06, 0.1, 0.2\}$. The pseudo-code for this model is provided in Algorithm 6 in Appendix A.

- **DH-MNN**: with piecewise linear basis functions and time knots equally distributed in intervals of 2. The pseudo-code for this model is provided in Algorithm 7 in Appendix A.

- **Cox**: the proportional hazard model [2] with the baseline hazard function being estimated using the Kalbfleish & Prentice estimator [11]. Competing risks were accounted for as in [3]. The pseudo-code for this model is provided in Algorithm 8 in Appendix A.

- **QR**: the quantile regression model, as in [26]. The pseudo-code for this model is provided in Algorithm 9 in Appendix A.

- **DeepSurv**: a neural network adaptation of the Cox model [13], which is equivalent to a restricted version of the PH-MNN model with a single time constant basis function. The pseudo-code for this model is provided in Algorithm 10 in Appendix A.

- **Cox-Time**: a neural network adaptation of the Cox model [16], which extends DeepSurv by the inclusion of time as one of the input variables, achieving universal approximation at the cost of an extremely increased computational cost. The pseudo-code for this model is provided in Algorithm 11 in Appendix A.

- **Cox-Time (fast)**: equal to Cox-Time but with less training iterations to make training time similar to other models. The pseudo-code for this model is provided in Algorithm 11 in Appendix A.

- **DeepHit**: a discrete time-interval model proposed in [45], which can be viewed as a direct hazard model. Two different time discretization intervals of 2 and 0.1 were used to study the effect of a large discretization interval on the model. Being a discrete time model, the conversion between cumulative incidence function and cause-specific representations is only fully specified at the limit for an infinitely small discretization step. This might

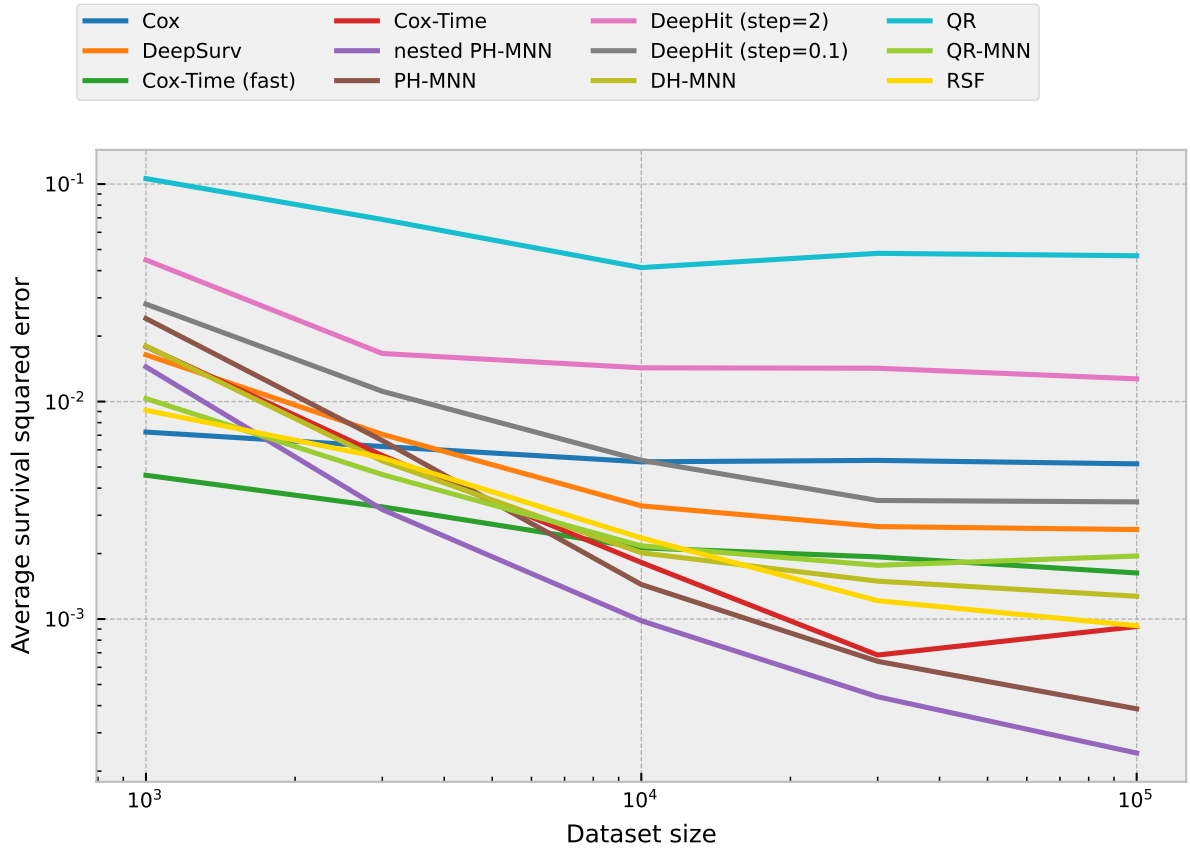


Figure 4.2.: Averaged integrated squared error of the survival function for different dataset sizes. The results are the average of 3 independent models trained with independently generated datasets.

lead to a greater estimation error when a large discretization step is used. The pseudo-code for this model is provided in Algorithm 12 in Appendix A.

- **RSF**: the random survival forest model implemented in scikit-survival [80]. The competing risks were taken into account by training two models separately and considering events other than the target one as censoring. With this procedure, each model will estimate the cause-specific hazard function for a different event type.

Fig. 4.2 shows how the averaged integrated squared error of the survival function varies with training dataset size. This squared error is defined as the integral over time of the squared difference between the true cumulative hazard function and the model estimation divided by the length of the time interval. Given that this is a synthetic data experiment, the underlying cumulative hazard function is known, allowing the estimation error of the model to be com-

puted. All of the MNN models performed better than previous state of the art in their respective model class. Despite most models reaching a saturation point where the error ceases to improve at the same rate as a function of the dataset size, the PH-MNN and nested PH-MNN maintain a clear trend of improvement in the entire range. This shows that the PH-MNN has more flexibility than the other models when given the same number of parameters, consistent with it's use of a nonparametric baseline hazard function. Fig. 4.3 shows how the averaged integrated squared error of the survival function evolves with model training time. Although in neural networks the training time is flexible and comparing training times of algorithms can be misleading, Figure 4.3 shows that all the proposed metaparametric neural network models have a shorter convergence curve than their respective existing state-of-the-art models. This means that the improvement achieved by the proposed models does not require a higher computational time to be achieved. Fig. 4.4 shows how each model es-

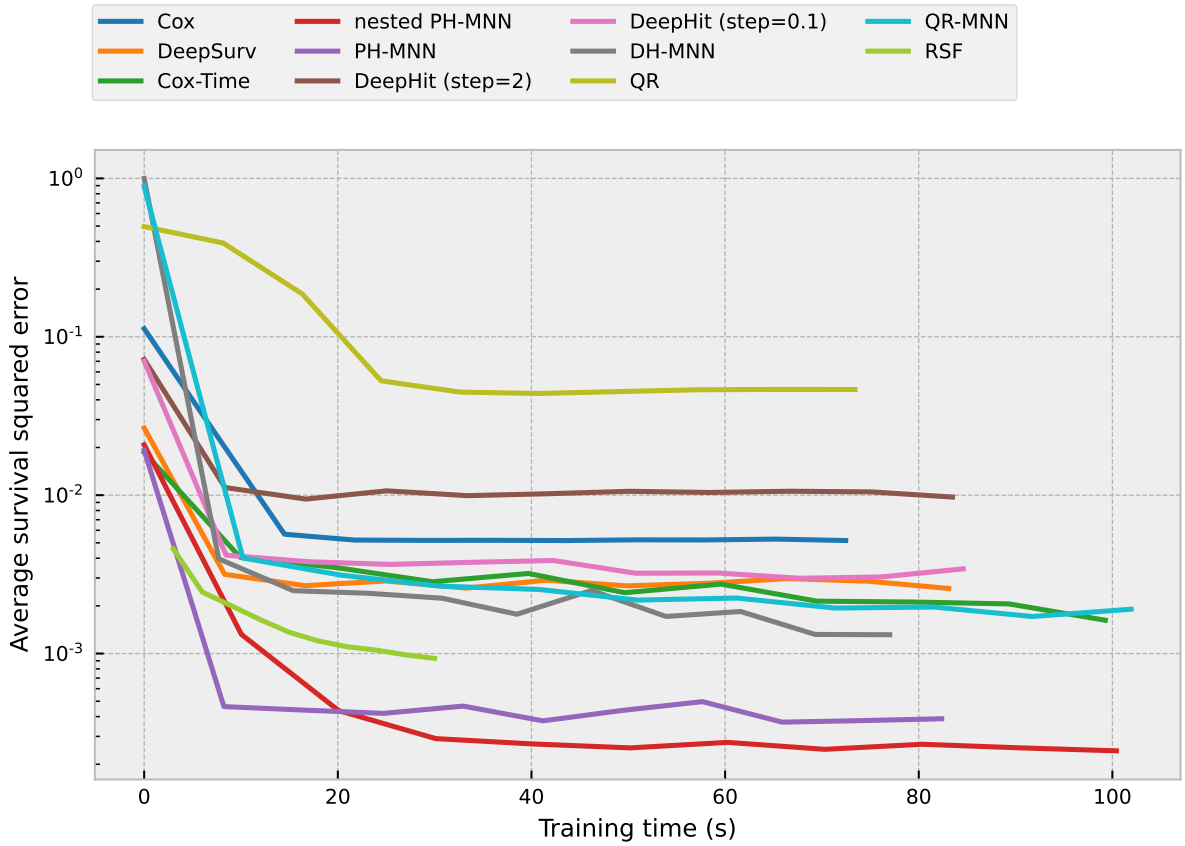


Figure 4.3.: Averaged integrated squared error of the survival function over training time using a single synthetic dataset with 100000 data points. Computation was performed with a RTX 2070 graphics card and the models were implemented with TensorFlow 2.3.1.

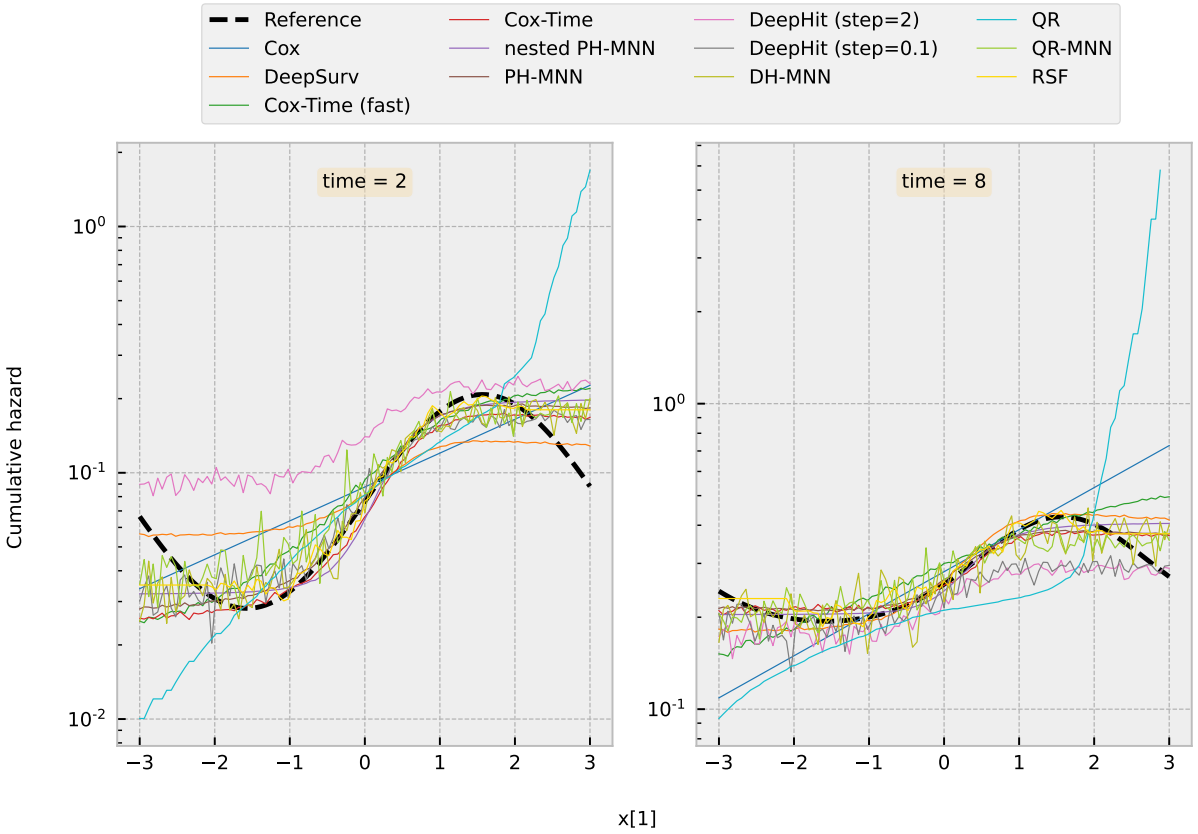


Figure 4.4.: Cumulative cause-specific hazard function for event type 1 in the synthetic data as a function of time and variable $x[1]$ when $x[0] = 0$. The values estimated are the averaged over 3 independent models trained with independently generated datasets, each one with 100000 data points.

timates the event type 1 in the synthetic data with $x[0] = 0$ as a function of time and $x[1]$. Note that all MNN models and also the DeepHit model are capable of representing the nonlinearities and time-dependencies in the model with different accuracies, as measured in Figs. 4.2 and 4.3. However, the DeepSurv, Cox and QR models are incapable of fully representing the target probability distribution, and so they would never converge to the underlying true probability distribution.

4.4.2 Application to a clinical dataset

We now proceed to the application of the proposed models to the estimation of the risks of death and revision surgery for patients who undergo hip replacement surgery, using data collected by the National Joint Registry in the United Kingdom. This dataset contains outcomes information from 1132875 hip replace-

ment surgeries performed from 2003 to 2019. Here, modeling was restricted to procedures performed from April 2009 to March 2019. Within this period, 855044 hip replacements were performed. The data was filtered to include only surgeries with complete data and only those where the reason for surgery was osteoarthritis, resulting in a total of 612914 procedures. The covariates used for estimation were:

- **numerical variables**:
  - **age**: restricted to the interval from 30 to 100;
  - **BMI**: restricted to the interval from 15 to 60;
  - **cup size:** ranging from 26 to 62;
  - **head size**: ranging from 26 to 60;
  - **stem size**: ranging from 26 to 46.
- **Boolean variables**:
  - **procedure type**: either complex or not;
  - **gender**: either male or female;
  - **bone graft acetabular**: whether or not this technique was used;
  - **bone graft femur**: whether or not this technique was used;
  - **minimally invasive technique**: whether or not this technique was used.
- **categorical variables**:
  - **Approach**: anterior, antero-lateral, direct anterior, Hardinge, Hardinge/anterolateral, lateral (inc Hardinge), posterior, trochanteric osteotomy or other;
  - **cup fixation**: cemented, cementless HA coated, cementless non HA coated;
  - **stem fixation**: cemented, cementless HA coated, cementless non HA coated;

- **side**: left, right or bilateral;
- **patient procedure**: cemented, cementless, other;
- **ASA**: American Society of Anaesthesiologists physical status classification ranging from 1 to 4;
- **primary surgical unit**: id of each of the 485 different surgical units from where hip replacements were registered;
- **implant reason**: one of 28 different medical indications for hip replacement;
- **cup type**: custom, monobloc, preassembled cup/liner, resurfacing, standard or not used;
- **cup composition**: ceramic, metal, metal ceramic combination, metal polyethylene combination, polyethylene, not used;
- **head type**: custom, modular, resurfacing or not used;
- **head composition**: ceramic, metal or not used.

To improve estimation performance, all numerical inputs were normalized through a linear transformation to make their mean 0 and their standard deviation 1. All categorical inputs were provided with one-hot encoding (i.e. a vector of c numbers where c is the number of categories and all values are equal to 0 except for the number that corresponds to the category specified, which is 1). As described in Figure 4.1, all categorical inputs passed through a layer with 10 hidden units, signoid activation and 0.7 Gaussian dropout.

The observed population survival curves are shown in the Kaplan-Meier estimate [6]. The performance of the proposed MNN models, together with those of benchmark and current state-of-the-art approaches were compared against the observed Kaplan-Meier estimate. The models used for comparison were the same as in Section 4.4.1. For the PH-MNN, the time knots used were 2, 4, and 7. For the DH-MNN and DeepHit, time knots were equally distributed in intervals of 6 months.

The models were evaluated with plots of the estimated cumulative hazard ratio (CHR) marginalized as a function of age and BMI. We chose age and BMI as

example predictor variables as they demonstrate a nonlinear relationship with survival, which the proposed methods should be able to capture. The marginalized CHR estimation as a function of either the age or the BMI used a sliding window with width equal to 4 in respective units and centered successively in each target value, where:

- the Kaplan-Meier estimate of the survival function within the window was performed, $S_{KM}(t|x \in \xi(w))$, where $\xi(w)$ is a window centered in w;
- the marginal model estimate within the window is computed as the average of the estimated survival function for each patient within the window, $S_{model}(t|x \in \xi(w))$;
- the Kaplan-Meier estimate was computed for the entire test population, $S_{KM}(t)$;
- the Kaplan-Meier estimation of the marginalized CHR was given by:

$$\mathrm{CHR_{KM}} = \frac{\log(S_{KM}(t|x \in \xi(w)))}{\log(S_{KM}(t))}; \tag{4.19}$$

- the model estimation of the marginalized CHR was given by:

$$\mathrm{CHR_{model}} = \frac{\log(S_{model}(t|x \in \xi(w)))}{\log(S_{KM}(t))}. \tag{4.20}$$

This process was repeated 250 times, for each model in a group of 50 random repetitions of 5-fold cross validation. The results of the estimated marginal CHR as a function of age or BMI averaged for all 250 runs are shown in Figs. 4.6-4.9. The results are evaluated according to the accuracy of representation of nonlinearities, adaptability of the shape as a function of time, and calibration. These three aspects are captured by the root mean square error of the model estimate of the log marginal CHR relative to the Kaplan-Meier estimate of the same quantity. For a given time t, this RMSE is given by:

$$RMSE = \frac{\int p(x \in \xi(w)) \log^2 \left(\frac{\log(S_{model}(t|x \in \xi(w)))}{\log(S_{KM}(t|x \in \xi(w)))} \right) \mathrm{d}w}{\int p(x \in \xi(w)) \mathrm{d}w}, \tag{4.21}$$

where $p(x)$ is the population density inside the window $\xi(w)$ centered in w, and w is either the age or the BMI. This RMSE represents an integrated measure of two

factors: first, the difference between the relationship of the model estimate and the observed data as a function of the attribute; and second the systematic bias between the two that is common for all values of the attribute. This bias can be defined at each time as the constant that if added to the estimation of the cumulative hazard ratio in that time would minimize the RMSE of the estimation as a function of either age or BMI. By estimating a bias that will minimize this RMSE, the two components can be identified as the unbiased RMSE (URMSE) and the bias. There, the unbiased RMSE is the residual RMSE in the estimation when the bias is removed. An intuition of the meaning of these error measures can be obtained by examining Figure 4.6. At time=0.25 years, it is possible to see that the estimation of the DeepHit model has a large positive bias, with the estimation being larger than the Kaplan-Meier estimation for most values of age. However, the shape of curve is similar to the Kaplan-Meier estimation, which means that the unbiased RMSE is small.

The model were evaluated through the computation of the RMSE, URMSE and absolute bias in the time interval from 6 months to 8 years with steps of 1 month. Tables 4.1, 4.2 and 4.3 present for each type of model the maximum value over time of each evaluation criteria with their 95% confidence interval. To highlight the improvement achieved with the metaparametric neural network structure,

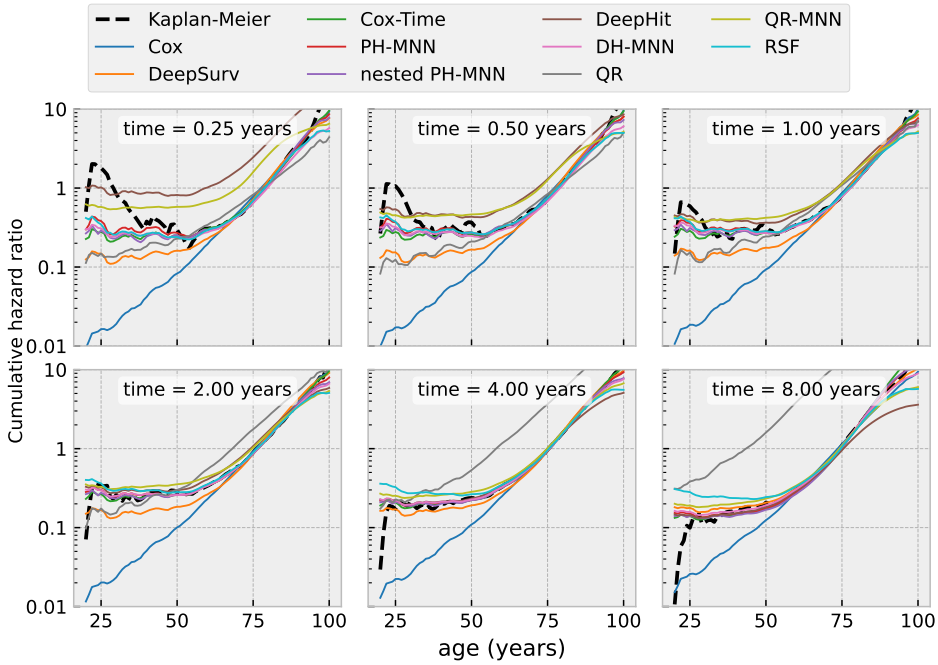


Figure 4.5.: Estimated cumulative hazard ratio for the mortality risk marginalized as a function of the age.

the results were grouped by type of model. For the direct hazards models, evaluation was performed in 6 months intervals to allow a fair comparison between both the discrete and continuous-time models.

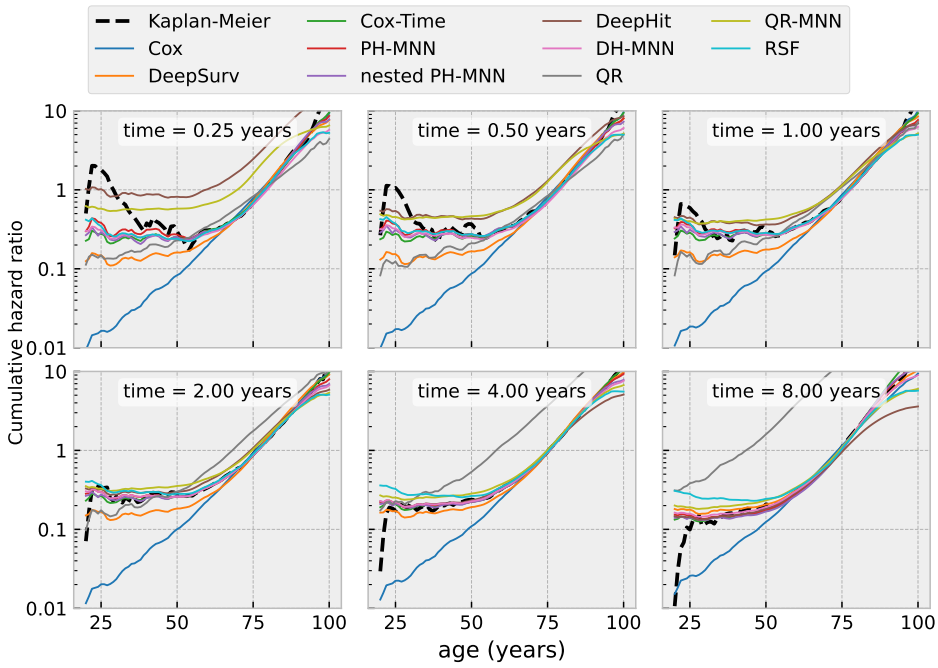


Figure 4.6.: Estimated cumulative hazard ratio for the mortality risk marginalized as a function of the age.

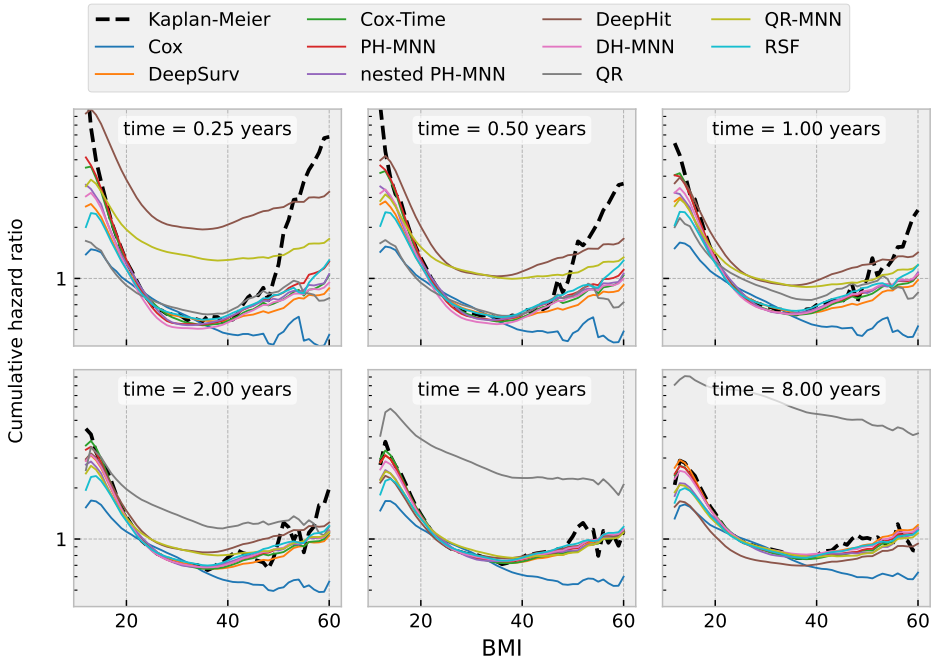


Figure 4.7.: Estimated cumulative hazard ratio for the mortality risk marginalized as a function of the BMI.

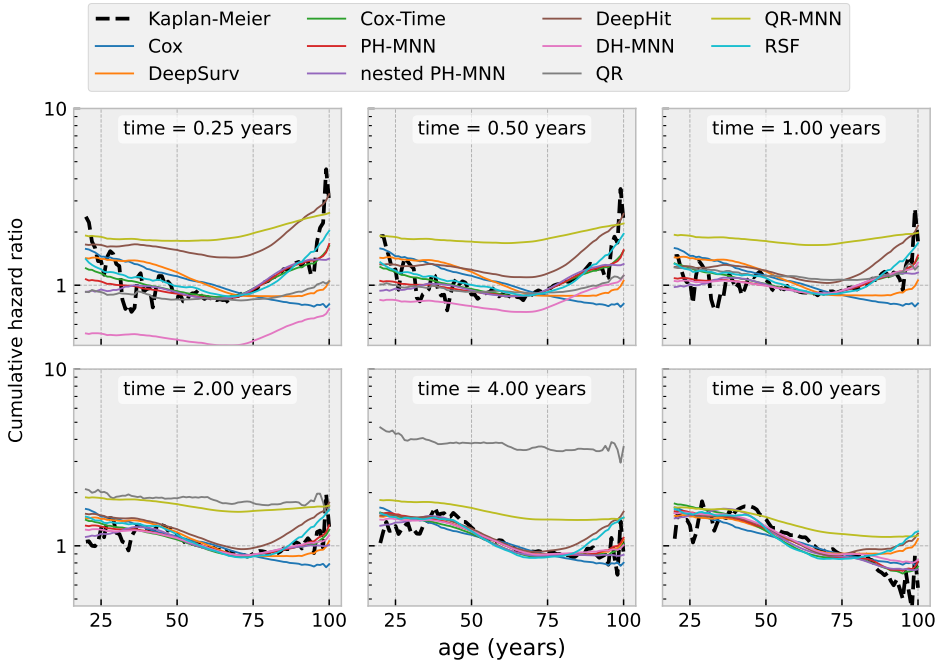


Figure 4.8.: Estimated cumulative hazard ratio for the revision risk marginalized as a function of the age.

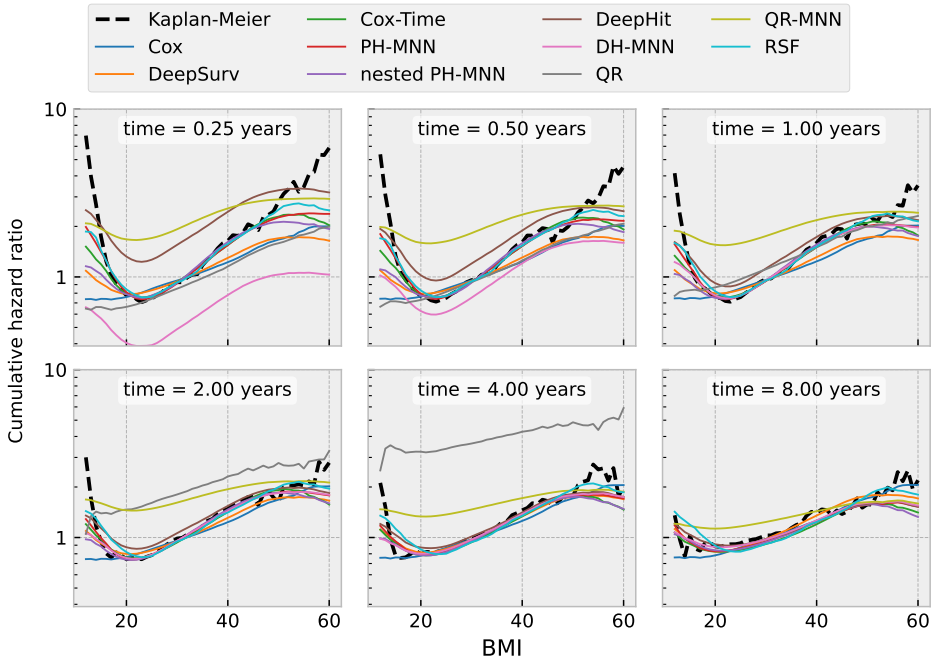


Figure 4.9.: Estimated cumulative hazard ratio for the revision risk marginalized as a function of the BMI.

Table 4.1.: Maximum value over time of each error component in proportional hazards models

		Cox	DeepSurv	Cox-Time	PH-MNN	nested PH-MNN	RSF
Revision by Age	RMSE	0.231 ± 0.010	0.217 ± 0.014	0.175 ± 0.004	$\mathbf{0.154 \pm 0.011}$	0.163 ± 0.010	0.174 ± 0.007
	URMSE	0.227 ± 0.013	0.214 ± 0.015	0.157 ± 0.005	$\mathbf{0.148 \pm 0.014}$	0.160 ± 0.011	0.169 ± 0.009
	abs. bias	0.059 ± 0.008	$\mathbf{0.054 \pm 0.007}$	0.097 ± 0.013	0.067 ± 0.001	0.065 ± 0.008	0.068 ± 0.005
Mortality by Age	RMSE	0.564 ± 0.005	0.310 ± 0.023	0.211 ± 0.027	$\mathbf{0.194 \pm 0.016}$	0.195 ± 0.028	0.199 ± 0.026
	URMSE	0.521 ± 0.007	0.281 ± 0.021	0.193 ± 0.025	0.189 ± 0.013	$\mathbf{0.188 \pm 0.026}$	0.192 ± 0.024
	abs. bias	0.236 ± 0.002	0.152 ± 0.016	0.099 ± 0.013	$\mathbf{0.057 \pm 0.011}$	0.073 ± 0.001	0.069 ± 0.010
Revision by BMI	RMSE	0.143 ± 0.003	0.137 ± 0.009	0.138 ± 0.017	0.122 ± 0.008	$\mathbf{0.118 \pm 0.005}$	0.121 ± 0.001
	URMSE	0.136 ± 0.003	0.130 ± 0.007	0.106 ± 0.001	$\mathbf{0.106 \pm 0.010}$	0.109 ± 0.003	0.107 ± 0.001
	abs. bias	0.071 ± 0.002	$\mathbf{0.063 \pm 0.000}$	0.101 ± 0.015	0.078 ± 0.002	0.066 ± 0.006	0.070 ± 0.004
Mortality by BMI	RMSE	0.192 ± 0.002	0.134 ± 0.002	0.147 ± 0.015	$\mathbf{0.132 \pm 0.005}$	0.133 ± 0.009	0.135 ± 0.007
	URMSE	0.187 ± 0.001	0.125 ± 0.004	0.122 ± 0.007	$\mathbf{0.119 \pm 0.007}$	0.126 ± 0.008	0.128 ± 0.006
	abs. bias	0.053 ± 0.007	0.054 ± 0.007	0.088 ± 0.013	0.062 ± 0.009	0.049 ± 0.004	$\mathbf{0.048 \pm 0.003}$

Table 4.2.: Maximum value over time of each error component in direct hazards models

		DeepHit	DH-MNN
Revision by Age	RMSE	0.463 ± 0.010	$\mathbf{0.260 \pm 0.018}$
	URMSE	0.161 ± 0.011	$\mathbf{0.148 \pm 0.013}$
	abs. bias	0.436 ± 0.004	$\mathbf{0.217 \pm 0.012}$
Mortality by Age	RMSE	1.006 ± 0.005	$\mathbf{0.242 \pm 0.033}$
	URMSE	0.198 ± 0.018	$\mathbf{0.186 \pm 0.015}$
	abs. bias	0.987 ± 0.007	$\mathbf{0.156 \pm 0.029}$
Revision by BMI	RMSE	0.449 ± 0.008	$\mathbf{0.241 \pm 0.015}$
	URMSE	0.111 ± 0.010	$\mathbf{0.105 \pm 0.010}$
	abs. bias	0.437 ± 0.005	$\mathbf{0.217 \pm 0.014}$
Mortality by BMI	RMSE	0.986 ± 0.011	$\mathbf{0.188 \pm 0.017}$
	URMSE	0.130 ± 0.004	$\mathbf{0.119 \pm 0.007}$
	abs. bias	0.977 ± 0.011	$\mathbf{0.146 \pm 0.019}$

The PH-MNN, DH-MNN, QR-MNN, DeepSurv and DeepHit models captured the nonlinearities, while the Cox did not. The QR model partially captured some nonlinearities through the variation of coefficients with the quantile, but they were not entirely captured since this variation is shared to represent both nonlinearities and time variations. This can be seen in the figures and is reflected by a smaller URMSE for the neural network models in most cases. The nonlinearities of the CHR could be adapted as a function of time for all the metaparametric neural networks and for the DeepHit model. The model structure for the others does not permit this variation of nonlinearities over time.

In the case of the proportional hazards models, the PH-MNN overall perform-

Table 4.3.: Maximum value over time of each error component in quantile regression models

		QR	QR-MNN
Revision by Age	RMSE	2.671 ± 0.077	$\mathbf{0.725 \pm 0.257}$
	URMSE	0.186 ± 0.014	$\mathbf{0.176 \pm 0.021}$
	abs. bias	2.666 ± 0.078	$\mathbf{0.700 \pm 0.259}$
Mortality by Age	RMSE	2.424 ± 0.125	$\mathbf{0.563 \pm 0.133}$
	URMSE	0.340 ± 0.008	$\mathbf{0.323 \pm 0.055}$
	abs. bias	2.403 ± 0.125	$\mathbf{0.448 \pm 0.110}$
Revision by BMI	RMSE	2.652 ± 0.080	$\mathbf{0.730 \pm 0.267}$
	URMSE	$\mathbf{0.149 \pm 0.014}$	0.172 ± 0.020
	abs. bias	2.649 ± 0.079	$\mathbf{0.710 \pm 0.265}$
Mortality by BMI	RMSE	2.012 ± 0.102	$\mathbf{0.496 \pm 0.082}$
	URMSE	$\mathbf{0.181 \pm 0.010}$	0.199 ± 0.025
	abs. bias	2.007 ± 0.102	$\mathbf{0.425 \pm 0.087}$

ance measured by the RMSE was better than the established methods. When this RMSE measure is broken down into its components, URMSE and absolute bias, the DeepSurv model had a slightly smaller bias. However the PH-MNN bias was still small and stable across the different risks and attributes. For the direct hazards models, the DH-MNN overall performance, URMS and bias were all consistently better than DeepHit. Finally, for the quantile regression models, the QR-MNN model overall performance was also consistently better than the QR method, apart from revision by BMI in which the two methods were equivalent.

4.5 Evaluation of MNN internal architecture

Now that we have shown the success of the MNN framework when compared to other survival models, we make a comparison between the baseline MNN architecture shown in the previous section with the nested MNN architecture proposed in Section 3.3.2.

4.5.1 Impact of hyper-parameters

In the first comparison, we use synthetically generated data to make an assessment of how the model hyper-parameters affect the estimation error in both the baseline MNN model and the nested MNN model. Data was generated from a proportional subdistribution hazard model with nonlinear and time-dependent

hazard ratio:

$$\tilde{\lambda}_1(t,\mathbf{x}) = 0.03(1+0.5\cos(2\pi t/10))\exp(\tan^{-1}(2x[0])\mathbb{1}(t<5)+\tan^{-1}(2x[1])\mathbb{1}(t>5))\,, \tag{4.22}$$

$$\tilde{\lambda}_2(t,\mathbf{x}) = 0.03(1+0.5\sin(2\pi t/10))\exp(\sin(x[1])\mathbb{1}(t<5)+\sin(x[0])\mathbb{1}(t>5))\,, \tag{4.23}$$

where $[x[0],x[1]] \sim \mathcal{N}([0,0],I_2)$. Censoring was included with uniform probability distribution for the censoring time within the interval $[0,10]$. Figure 4.10 shows the response of both the standard PH-MNN model and the nested mixture model PH-MNN to changes in the hyper-parameters. The the same dataset size is used for all analysis. In the standard PH-MNN model the overfit was a function of the number of hidden layers in the neural network, but in the nested mixture model PH-MNN it was a function of the number of knots in the explicit variable basis function representation. This improves interpretability of the choice of hyper-parameters in the model since the marginal distribution of the explicit variable can easily be observed from the raw data.

4.5.2 Nested PH-MNN mode analysis on COVID-19 hospitalization data

The Brazilian COVID-19 hospitalization dataset included data from all known COVID-19 related hospitalizations in Brazil from 1 Jan 2021 to 26 Nov 2021, with a total of more than 1,1 million hospitalizations. We have used this dataset to illustrate how the nested PH-MNN structure can be used to help interpretation and validation of the model. The data was filtered to include only patients who were hospitalized and confirmed with COVID-19. The outcome was split between mortality and hospital discharge. The following input variables were used in the model:

- **numerical variables**:
  - **age**: ranging from 0 to 100;
  - **hospitalization day**: day of the year when hospitalization happened ranging from 0 to 300.
- **Boolean variables**:
  - **Sex**: patient sex;

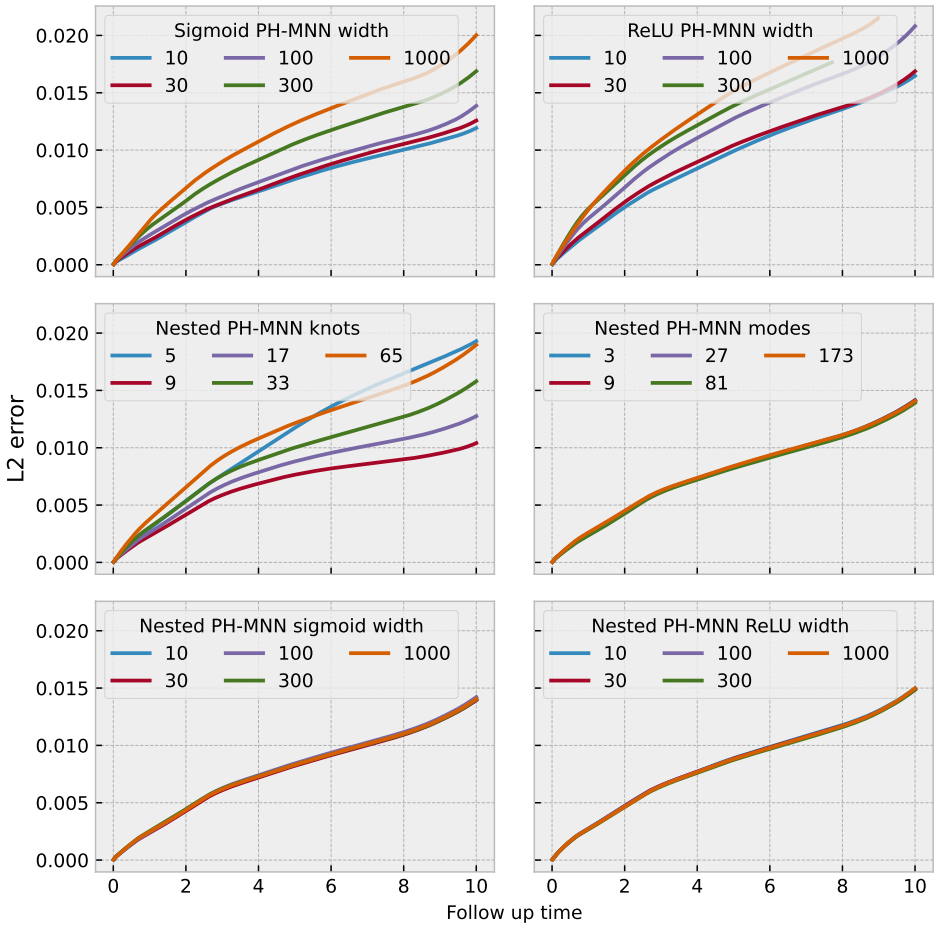
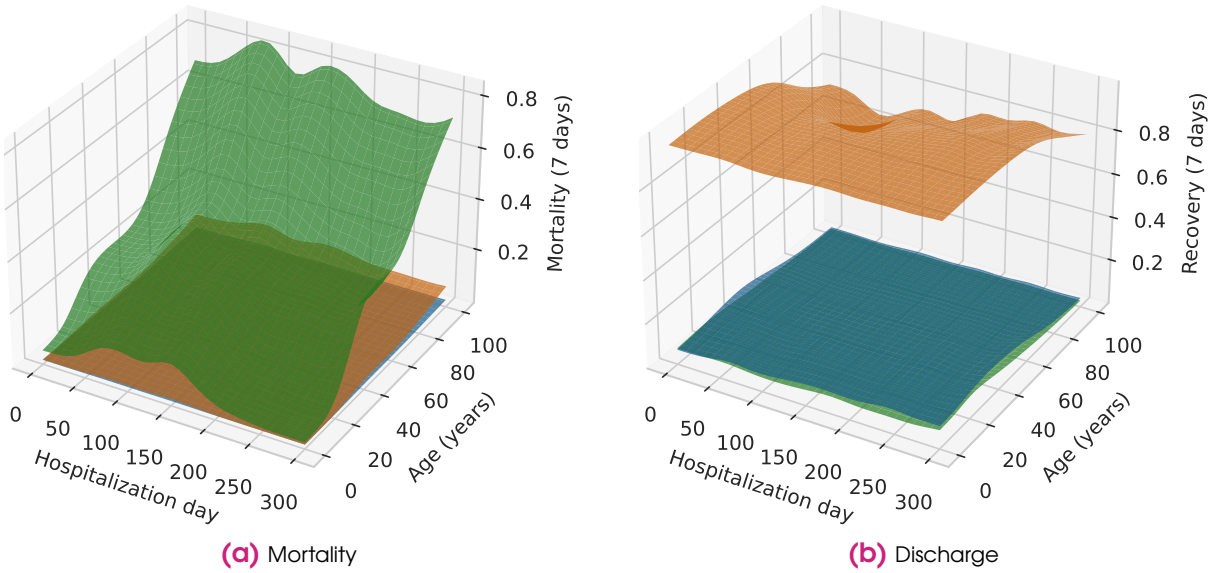


Figure 4.10.: L2 error of the cumulative incidence function as a function of various hyper-parameters for both standard and explicit PH-MNN models.



(a) Mortality **(b)** Discharge

Figure 4.11.: Mortality and discharge probabilities 7 days after hospitalization by COVID-19.

- **Hospital Contamination**: whether contamination happened at the hospital;
- **Risk - Puerperium**: whether patient has recently delivered a child;
- **Risk - Chronic cardiac disease**: whether or not any chronic cardiac disease is present;
- **Risk - Down syndrome**: whether or not the patient has Down syndrome;
- **Risk - Chronic liver disease**: whether or not any chronic liver disease is present;
- **Risk - Asthma**: whether or not patient suffers from asthma;
- **Risk - Diabetes mellitus**: whether or not patient suffers from diabetes;
- **Risk - Neurological disease**: whether or not patient suffers from any neurological disease;
- **Risk - Other chronic lung disease**: whether or not patient suffers from any chronic lung disease;
- **Risk - Immunodeficiency**: whether or not suppers from any type of immunodeficiency;
- **Risk - Chronic kidney disease**: whether or not any chronic kidney disease is present;
- **Risk - Obesity**: whether or not patient is obese;
- **Risk - Others**: whether or not other risks were present.

- **categorical variables**:
  - **Pregnancy**: 8 possible values distinguishing whether or not the patient is pregnant and the pregnancy period;
  - **Notification UF**: one of 28 Brazilian Federation Units;
  - **Risk - Other (description)**: 81636 different values of `Other' risk description;

- **Ethnicity**: white, black, mixed ethnic groups, yellow or indigenous;
- **City**: 2860 different cities from which data was available;
- **Healthcare Unit**: 5844 different healthcare units from which data was available.

The model was structured with the nested PH-MNN structure described in Figure 3.4 with a total of 3 modes. Time was represented with a piecewise linear basis functions and time knots 2,7,14,28 and 60 days. The patient's age and the hospitalization day were modeled as explicit variable using natural cubic splines with knots 0,30,60,90,120,150,180,210,240,270 and 300 for the hospitalization day and 0,10,20,40,60,80 and 100 for age. The remaining variables were used as input of a neural network with softmax output in order to compute the weight of each basis function in a mixture model using the structure as in Figure 4.1. Algorithm 1 provides the pseudo-code for the computation of the model outcome: Figure 4.11 show the three resulting basis function for the 7 days mortality

Algorithm 1 Computation of the hazard ratio (ω_j) in the nested PH-MNN model.

Require: a (age)
Require: d (hospitalization day)
Require: t (follow-up time)
Require: $\boldsymbol{\psi}(\boldsymbol{x})$ (output of the neural network block of the model)
Require: $\nu_k^{[d]}(d)$ (natural cubic splines basis functions for hospitalization day)
Require: $\nu_k^{[a]}(a)$ (natural cubic splines basis functions for age)
Require: $\nu_k^{[t]}(t)$ (piecewise linear basis functions for follow-up time)
Require: $\Theta_{1,j}^{4\times7\times11}$ (parameters)
Require: $\Theta_{2,j}^{3\times5\times4}$ (parameters)
  for j in event types **do**
    $e_{i,j}(d,a) \leftarrow \sum_{k,l} \Theta_{1,j}^{i,l,k} \nu_l^{[a]}(a) \nu_k^{[d]}(d)$
    $w_{m,n,j}(d,a) \leftarrow \mathrm{softmax}(\sum_i \Theta_{2,j}^{m,n,i} e_{i,j}(d,a))$
    $\omega_j(t,d,a,\boldsymbol{x}) \leftarrow \sum_{m,n} w_{m,n,j}(d,a) \nu_n^{[t]}(t) \boldsymbol{\psi}_m(\boldsymbol{x})$
  end for

and discharge respectively as a function of age and hospitalization day. This shows the age dependency is nearly constant over time and has approximately the same shape across all modes, but the total risk magnitude differs significantly depending on other covariates. Figure 4.12 shows the distribution of weights for the different modes, being an illustration of how variable other than age and hospitalization day affect the outcome.

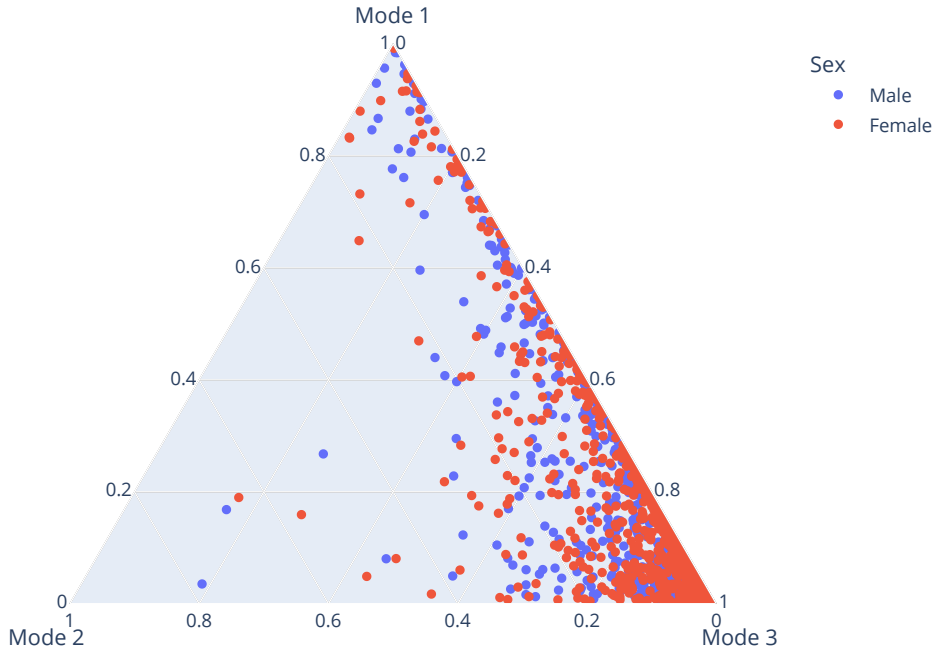


Figure 4.12.: Distribution of weights for each of the three modes among the population as a function of other input variables.

5

Maximum Likelihood Derivation of the PH-MNN Estimator and its Properties

In Chapter 4, the metaparametric neural network framework was successfully applied to survival analysis, leading an improvement over the state-of-the-art. This was done with three different models: proportional hazards (PH-MNN), direct hazard (DH-MNN) and quantile regression (QR-MNN). The experimental success is consistent with Theorem 1 which guarantees that the MNN framework can represent any continuous survival probability distribution. In special, the PH-MNN model achieved the best results among all three alternatives, which can be explained by the use of the baseline hazard function. Indeed, this use allows the model to simultaneously have a high time resolution for covariate independent components and a low time resolution for covariate dependent ones. Nonetheless, despite the vast literature on the theoretical justifications for the proportional hazards model, it does not cover all possible scenarios and the PH-MNN model is capable of representing patterns that are not covered by current literature. Although the experiments in Sections 4.4 and 4.5 show the success of the PH-MNN in a variety of scenarios, it is important to expand the theoretical background for the proportional hazards model in order to provide more confidence in the use of the PH-MNN model.

The two main aspects of a data based model that makes it adequate or not

to some estimation task are: the capability of representing the target patterns and the convergence of the model to the closest possible representation of the true pattern as the dataset increases. The adoption of a metaparametric neural network structure aims specifically at improving the first aspect. Indeed, the PH-MNN model can represent virtually any survival function according to Theorem 1, and Chapter 4 provides examples of patterns that cannot be represented by a standard proportional hazards model and are successfully represented by a PH-MNN model. Once it has been established that the universal approximation is achieved by the PH-MNN model, it is now necessary to show that the model convergence is not harmed by the introduction of additional complexity to the model. This chapter aims specifically at studying the estimation of a PH-MNN model considering the adequacy of the likelihood function and its convergence properties. Similarly to other neural network models, a proof of convergence would be challenging. Nonetheless, in the present chapter we provide a proof of convergence assuming that the hidden layers have been pre-trained and set to be constant during the training of the linear basis function layers. This does not mean that the layers in the MNN model cannot be trained jointly in practice. Indeed, it has already been experimentally demonstrated in Chapter 4 that this can be successfully done.

Current literature already provides analysis of the likelihood function and convergence properties for the standard proportional hazards model. Proofs of the asymptotic properties of the partial likelihood estimator in the standard Cox model were provided in [81] and [82]. Some methods allow the direct optimization of the total likelihood and they can be divided into two groups. The first group consists of methods in which a parametric model is used for the baseline hazard function, permitting joint optimization with the hazard ratio [22, 23]. The second group of methods estimate the baseline hazard function as a function of the hazard ratio, allowing a likelihood expression that depends exclusively on the hazard ratio. The resulting likelihood is called a profile likelihood. The modeling of the baseline hazard function as a piecewise constant function that only changes when an event is observed has been proposed by [10]. The profile likelihood in this model is proportional to the partial likelihood, which shows that optimizing the partial likelihood or the total likelihood is equivalent if the piecewise constant baseline hazard function is used. Another method was proposed by [11]. In this method, nonparametric estimation of the baseline hazard function is performed as a function of a time-invariant hazard ratio that relies on partial like-

lihood maximization. Although the Breslow and the Kalbfleisch & Prentice methods require a time-constant hazard ratio, the method in [83] does not. Here, the baseline cumulative hazard function is estimated within a counting process framework. A profile likelihood is obtained that is also proportional to the partial likelihood. Nevertheless, this specific counting process formulation allows multiple instances of the same event to be experienced simultaneously by the same subject, which is incompatible with conventional survival analysis problems. The first to propose a joint estimation method based on a profile likelihood that is not equivalent to the partial likelihood estimation was [84]. However, this method also assumes a time-invariant hazard ratio.

The competing risks extensions of the proportional hazards model has been performed in three different way in the literature: the cause specific hazard [3]; the subdistribution hazard [4]; and the mixture model [77, 85, 86]. In their standard formulation, each of the extensions will allow the representation of a particular subset of survival functions. Therefore, each formulation might be best suited to a different set of problems and it is difficult to know the problems in which each model will perform the best. In the PH-MNN extension, this problem becomes of little relevance since one single formulation can represent virtually any survival function. With this extended model flexibility, it is necessary to investigate whether the same convergence properties of standard competing models still hold. Indeed, all convergence analysis of competing risks models found in the literature were restricted to time-constant linear models and the PH-MNN can be at the same time both tide-dependent and nonlinear.

In Section 5.1, we propose a novel semi-parametric approach in which we first choose a generic covariate dependent survival function. Then, through nonparametric maximum likelihood estimation, a subset of the possible survival functions that includes the optimal model structure is obtained. This novel procedure creates a time-dependent and competing risks extension of the model proposed by [11] and [84]. It is proved that the supremum of the likelihood within the resultant subset of models is equal to the supremum of the likelihood within the class of all survival models. In most survival analysis applications it is not possible to perform an entirely nonparametric maximum likelihood estimation, which would only be achievable if for each possible combination of input variables, there were enough data to create a different Kaplan-Meier model. Therefore, an "a posteriori" parametric representation is required after performing the first step of nonparametric estimation. Nevertheless, the estimation procedure

is not affected by the parametrization chosen, thus allowing an approach that better suits any particular problem without the need to develop a new estimator. Indeed, in Section 5.1.6 we show that the Kaplan-Meier estimator and variations of the proportional hazards, the proportional subdistribution hazard, and the mixture model can be obtained as particular instances of the proposed model. The structure of the resulting model can be seen as the theoretical substract of a model without a particular implementation and the PH-MNN provides and implementation to it. In Section 5.2, it is shown that in a particular subset of PH-MNN models there is a class of objective functions with equivalent asymptotic behavior to which both the partial likelihood and the proposed profile likelihood belong. However, the small sample behavior of each estimator in this class is different. We additionally show that both the partial likelihood and the profile likelihood estimators will be biased for small samples, and propose an alternative objective function that compensates for this bias.

5.1 Maximum likelihood derivation for semi-parametric survival models

In the present section, we provide an alternative derivation for the PH-MNN model where nonparametric estimation is used to partially constrain a completely generic survival function. The model obtained by this procedure was named the coupled baseline hazard (CBH) model. It depends on a multidimensional function $\omega_j(\mathbf{x}, t)$ and provides a profile likelihood for estimation of $\omega_j(\cdot)$ and a procedure for obtained the survival function from $\omega_j(\cdot)$. Representing the function $\omega_j(\mathbf{x}, t)$ with an MNN with t as the explicit variable leads to the PH-MNN model, thus providing a mechanism for maximum likelihood estimation of the PH-MNN model without need of the partial likelihood.

5.1.1 Problem statement

In a competing risks survival model scenario, we assume that for a set of subjects in a dataset, the following data will be available:

- $\mathbf{x}_n$: the vector of covariates for subject n.
- T_n: time when subject n either stoped being observed or experienced one

of the events of interest.

- E_n: and indicator that is 1 when some event of interest has happened to subject n in time T_n and that is 0 when subject n stoped being observed at time T_n without experiencing any of the events of interest.

- j_n: is the type of event observed for subject n, which is undefined in the case of censoring.

We denote this dataset as follows:

$$\mathcal{D} = \{\mathbf{x}_n; T_n; j_n; E_n | n \in \{1, 2, \ldots, N\}; T_n \leq T_m, \forall n < m\}\,. \tag{5.1}$$

In this scenario, we first define function $N_n(t)$ which is 0 if subject n has not experienced any event of any type until time t and 1 otherwise. It is then possible to define the overall survival function as the probability that a subject will not experience any event type until a particular time t:

$$S(\mathbf{x}_n, t) = P(N_n(t) = 0 | \mathbf{x})\,. \tag{5.2}$$

This probabilistic description of the event time can be extended to the competing risks scenario by combining the overall survival function with the probability of each event type given the time when the event happened:

$$p_j(\mathbf{x}, t) = P(j_n = j | \mathbf{x}_n = \mathbf{x}, E_n = 1, T_n = t)\,. \tag{5.3}$$

We denote this set of probability distributions as follows:

$$\mathbf{S}(\mathbf{x}, t) = [S(\mathbf{x}, t), p_1(\mathbf{x}, t), \ldots, p_J(\mathbf{x}, t)]\,. \tag{5.4}$$

The likelihood of the dataset $\mathcal{D}$ for a given survival model $\mathbf{S}(\mathbf{x}, t)$ is given by:

$$\mathcal{L}\{\mathcal{D}|\mathbf{S}\} = \prod_{n=1}^{N} \mathcal{L}\{\mathcal{D}_n|\mathbf{S}\}\,, \tag{5.5}$$

where

$$\mathcal{L}\{\mathcal{D}_n|\mathbf{S}\} = \begin{cases} S\left(\mathbf{x}_n, T_n\right) & , E_n = 0\,, \\ \left[S\left(\mathbf{x}_n, T_n^-\right) - S\left(\mathbf{x}_n, T_n\right)\right] p_{j_n}\left(\mathbf{x}_n, T_n\right) & , E_n = 1\,. \end{cases} \tag{5.6}$$

An intuition of the meaning of each component in equation (5.6) can be

achieved by analysis each case separately. If $E_n = 0$, all the information contained in the dataset about subject n apart from the input covariates $\mathbf{x}_n$ is that no event has been observed until time T_n, whose probability is given by the overall survival function $S(\mathbf{x}_n, T_n)$. If $E_n = 1$, then event j_n has happened at time T_n and no other event has happened before. Denoting $S(\mathbf{x}_n, T_n^-) = \lim_{t \to T_n^-} S(\mathbf{x}_n, t)$, we have that $S(\mathbf{x}_n, T_n) = P(N_n(T_n) = 0 | \mathbf{x}_n)$ and $S(\mathbf{x}_n, T_n^-) = P(N_n(T_n^-) = 0 | \mathbf{x}_n)$. Therefore, $S(\mathbf{x}_n, T_n^-) - S(\mathbf{x}_n, T_n) = P(N_n(T_n^-) = 0, N_n(T_n) = 1 | \mathbf{x}_n)$, which is the probability of the first event happening at time T_n, regardless of its type. Thus, the probability of the first event happening at time T_n with event type j_n is given by $\left[S\left(\mathbf{x}_n, T_n^-\right) - S\left(\mathbf{x}_n, T_n\right)\right] p_{j_n}\left(\mathbf{x}_n, T_n\right)$.

When estimating a function through nonparametric maximum likelihood estimation, the target is to find the function that will maximize the likelihood of the data given the model. This can be mathematically formulated as finding $\mathbf{S}^*(\mathbf{x}, t) = \arg\max_{\mathcal{S}}(\mathcal{L}\{\mathcal{D}_n | \mathbf{S}\})$, where $\mathcal{S}$ is the class of all possible survival models. This optimization is performed in the class of all possible functions in a particular domain. Therefore, the dimensionality of the search space in this optimization problem is infinite, in contrast with the case of parametric maximum likelihood estimation where the search space of the target function is limited to a finite dimensionality through a parametrization where a finite number of parameters is used. As detailed in Section 5.1.5, we only partially constrain the model using nonparametric maximum likelihood estimation as a purely nonparametric estimation procedure would be unfeasible due to the required dataset size that should grow exponentially with the number of input covariates in the model. Instead, we impose successive restrictions to the nonparametric model $\mathbf{S}(\mathbf{x}_n, t)$ where in every step it is guaranteed that the supremum of the likelihood in the restricted search space is equal to the supremum of the likelihood in the broader search space. This task can be formulated mathematically as finding a subset of the class of all possible survival models $\mathcal{C} \subset \mathcal{S}$ so that $\sup_{\mathcal{C}}(\mathcal{L}\{\mathcal{D}_n | \mathbf{S}\}) = \sup_{\mathcal{S}}(\mathcal{L}\{\mathcal{D}_n | \mathbf{S}\})$.

5.1.2 Piecewise constant restriction of the model

In [6], it was proven that the Kaplan-Meier model is the maximum likelihood estimator for a generic covariate-independent survival scenario with random censoring. This proof can be divided into three steps:

Step 1. *showing that the maximum likelihood estimator for the survival function will necessarily be a piecewise constant model in which the survival function only changes when an event occurs.*

Step 2. *rewriting the likelihood function in a factorized form so that each variable to be estimated appears in a separate factor.*

Step 3. *analytically optimizing the factorized likelihood to obtain a closed form solution.*

In this section, we extend steps 1 and 2 to a generic survival model with competing risks and covariate dependent survival function. We use the following notation: $\mathbf{x}_n$ is the vector of covariates for subject n; T_n is the minimum observed time for subject n (either event or censoring); j_n is the type of event observed for subject n, which is undefined in the case of censoring; E_n is the indicator of censoring, which is 0 if subject n is censored and 1 if uncensored; $S(\mathbf{x}, t)$ is the overall survival function; $p_j(\mathbf{x}, t)$ is the probability of event type j given that an event occurred at time t.

Lemma 3. *Given: a dataset* $\mathcal{D} = \{\mathbf{x}_n; T_n; j_n; E_n | n \in \{1, 2, \ldots, N\}; T_n \leq T_m, \forall n < m\}$*; a competing risks survival model* $\mathbf{S}(\mathbf{x}, t) = [S(\mathbf{x}, t), p_1(\mathbf{x}, t), \ldots, p_J(\mathbf{x}, t)]$*; and a piecewise constant version of* $\mathbf{S}(\mathbf{x}, t)$ *denoted by* $\mathbf{S}^*(\mathbf{x}, t) = [S^*(\mathbf{x}, t), p_1(\mathbf{x}, t), \ldots, p_J(\mathbf{x}, t)]$ *so that:* $S^*(\mathbf{x}, t) = 1$ *if* $t < T_1$*; and* $\mathbf{S}^*(\mathbf{x}, t) = \mathbf{S}(\mathbf{x}, T_n)$ *otherwise, where* $n = \max\{n | T_n \leq t\}$*. The likelihood of* $\mathbf{S}^*(\mathbf{x}, t)$ *is greater or equal to the likelihood of* $\mathbf{S}(\mathbf{x}, t)$*.*

Proof. The likelihood of $\mathbf{S}(\mathbf{x}, t)$ is given by:

$$\mathcal{L}\{\mathcal{D}|\mathbf{S}\} = \prod_{n=1}^{N} \mathcal{L}\{\mathcal{D}_n|\mathbf{S}\}\,, \tag{5.7}$$

where

$$\mathcal{L}\{\mathcal{D}_n|\mathbf{S}\} = S\left(\mathbf{x}_n, T_n\right)^{(1-E_n)} \left[S\left(\mathbf{x}_n, T_n^-\right) - S\left(\mathbf{x}_n, T_n\right)\right]^{E_n} p_{j_n}\left(\mathbf{x}_n, T_n\right)^{E_n}\,. \tag{5.8}$$

Analysing separately $\mathcal{L}\{\mathcal{D}_n|\mathbf{S}\}$ for the possible values of E_n, it can be shown that:

- For $E_n = 0$:

$$\mathcal{L}\{\mathcal{D}_n|\mathbf{S}\} = S\left(\mathbf{x}_n, T_n\right) = S^*\left(\mathbf{x}_n, T_n\right) = \mathcal{L}\{\mathcal{D}_n|\mathbf{S}^*\}\,. \tag{5.9}$$

- For $E_n = 1$:

$$\mathcal{L}\{\mathcal{D}_n|\mathbf{S}\} = \left[S\left(\mathbf{x}_n, T_n^-\right) - S\left(\mathbf{x}_n, T_n\right)\right] p_{j_n}\left(\mathbf{x}_n, T_n\right) \leq \left[S\left(\mathbf{x}_n, T_{\nu[n]-1}\right) - S\left(\mathbf{x}_n, T_n\right)\right] p_{j_n}\left(\mathbf{x}_n, T_n\right) = \mathcal{L}\{\mathcal{D}_n|\mathbf{S}^*\}, \tag{5.10}$$

  where $\nu[n] = \min\{\nu \in \{1, \ldots, N\} | T_\nu = T_n\}$.

Thus, $\mathcal{L}\{\mathcal{D}_n|\mathbf{S}^*\} \geq \mathcal{L}\{\mathcal{D}_n|\mathbf{S}\}, \forall n \in \{1, \ldots, N\} \Rightarrow \mathcal{L}\{\mathcal{D}|\mathbf{S}^*\} = \prod_{n=1}^{N} \mathcal{L}\{\mathcal{D}_n|\mathbf{S}^*\} \geq \prod_{n=1}^{N} \mathcal{L}\{\mathcal{D}_n|\mathbf{S}\} = \mathcal{L}\{\mathcal{D}|\mathbf{S}\}$. □

Lemma 4. *Given a dataset* $\mathcal{D} = \{\mathbf{x}_n; T_n; j_n; E_n | n \in \{1, 2, \ldots, N\}; T_n \leq T_m, \forall n < m\}$ *and a piecewise constant survival model* $\mathbf{S}(\mathbf{x}, t) = [S(\mathbf{x}, t), p_1(\mathbf{x}, t), \ldots, p_J(\mathbf{x}, t)]$ *so that:* $S(\mathbf{x}, t) = 1$ *if* $t < T_1$*; and* $\mathbf{S}(\mathbf{x}, t) = \mathbf{S}(\mathbf{x}, T_n)$ *otherwise, where* $n = \max\{n | T_n \leq t\}$*. It is possible to represent* $\mathbf{S}(\mathbf{x}, t)$ *through a custom set of functions* $f_{j,n}(x)$*, with the following properties:*

Property 1. $S(\mathbf{x}, T_n) = \prod_{m=1}^{n} \left[1 - \sum_{j=1}^{J} f_{j,m}(\mathbf{x})\right]$.

Property 2. $p_j(\mathbf{x}, T_n) \sum_{\gamma=1}^{J} f_{\gamma,n}(\mathbf{x}) = f_{j,n}(\mathbf{x})$.

Property 3. $f_{j,m}(\mathbf{x}) \geq 0$, $\sum_{j=1}^{J} f_{j,m}(\mathbf{x}) \leq 1, \forall \mathbf{x}, j$.

The likelihood of $\mathbf{S}(\mathbf{x}, t)$ *will be given by* $\mathcal{L}\{\mathcal{D}|\mathbf{S}\} = \prod_{n=1}^{N} \mathcal{L}_n$*, where:*

$$\mathcal{L}_n = \begin{cases} \prod_{m=n}^{N} \left[1 - \sum_{j=1}^{J} f_{j,n}(\mathbf{x}_m)\right] & , E_n = 0, \\ f_{j_n,\nu[n]}(\mathbf{x}_n) \prod_{m=\nu[n]+1}^{N} \left[1 - \sum_{j=1}^{J} f_{j,\nu[n]}(\mathbf{x}_m)\right] & , E_n = 1, \end{cases} \tag{5.11}$$

and $\nu[n] = \min\{\nu \in \{1, \ldots, N\} | T_\nu = T_n\}$.

Proof. Making $f_{j,n}(\mathbf{x}) = p_j(\mathbf{x}, T_n)\left[1 - s_n(\mathbf{x})\right]$, where: $s_n(\mathbf{x}) = 0$, if $S(\mathbf{x}, T_{n-1}) = 0$; and $s_n(\mathbf{x}) = S(\mathbf{x}, T_n)/S(\mathbf{x}, T_{n-1})$ otherwise. $S(\mathbf{x}, T_n)$ will be fully specified for all $\mathbf{x}$ and t; and $p_j(\mathbf{x}, T_n)$ will be fully specified in points where $S(\mathbf{x}, t) < \lim_{t \to t^-} S(\mathbf{x}, t)$. The properties are obtained as follows:

- property 1:

$$\sum_{j=1}^{J} p_{j,n}(\mathbf{x}) = 1 \Rightarrow \sum_{j=1}^{J} f_{j,n}(\mathbf{x}) = 1 - s_n(\mathbf{x}) \Rightarrow S(\mathbf{x}, T_n) = \prod_{m=1}^{n} \left[1 - \sum_{j=1}^{J} f_{j,m}(\mathbf{x})\right], \tag{5.12}$$

- property 2:

$$p_j(\mathbf{x}, T_n) \sum_{\gamma=1}^{J} f_{\gamma,n}(\mathbf{x}) = p_j(\mathbf{x}, T_n)(1 - s_n(\mathbf{x})) = f_{j,n}(\mathbf{x}) \,, \tag{5.13}$$

- property 3 (if $S(\mathbf{x}, T_{n-1}) = 0$):

$$s_n(\mathbf{x}) = 0 \,, \tag{5.14}$$

- property 3 (other cases):

$$0 \leq S(\mathbf{x}, T_n) \leq S(\mathbf{x}, T_{n-1}) \leq 1 \Rightarrow 0 \leq s_n(\mathbf{x}) \leq 1 \,. \tag{5.15}$$

  Additionally,

$$0 \leq p_{j,n}(\mathbf{x}) \leq 1 \,. \tag{5.16}$$

The likelihood of $\mathbf{S}$ can be computed as follows:

$$\begin{aligned}
\mathcal{L}\{\mathcal{D}|\mathbf{S}\} &= \prod_{n=1}^{N} S\left(\mathbf{x}_n, T_{\nu[n]}\right)^{(1-E_n)} \left[\left[S\left(\mathbf{x}_n, T_{\nu[n]-1}\right) - S\left(\mathbf{x}_n, T_{\nu[n]}\right)\right] p_{j_n,n}\left(\mathbf{x}_n\right)\right]^{E_n} \\
&= \left[\prod_{n=1}^{N} \prod_{k=1}^{\nu[n]-1} s_k(\mathbf{x}_n)\right] \prod_{n=1}^{N} s_{\nu[n]}\left(\mathbf{x}_n\right)^{(1-E_n)} f_{j_n,\nu[n]}\left(\mathbf{x}_n\right)^{E_n} \\
&= \left[\prod_{n=1}^{N} \prod_{k=\nu[n]+1}^{N} s_n\left(\mathbf{x}_k\right)\right] \prod_{n=1}^{N} s_{\nu[n]}\left(\mathbf{x}_n\right)^{(1-E_n)} f_{j_n,\nu[n]}\left(\mathbf{x}_n\right)^{E_n} \\
&= \prod_{n=1}^{N} \left[s_{\nu[n]}\left(\mathbf{x}_n\right)^{(1-E_n)} f_{j_n,\nu[n]}\left(\mathbf{x}_n\right)^{E_n} \prod_{k=\nu[n]+1}^{N} s_n\left(\mathbf{x}_k\right) \right] \\
&= \prod_{n=1}^{N} \left[\left[1 - \sum_{j=1}^{J} f_{j,\nu[n]}\left(\mathbf{x}_n\right)\right]^{(1-E_n)} f_{j_n,\nu[n]}\left(\mathbf{x}_n\right)^{E_n} \prod_{k=\nu[n]+1}^{N} 1 - \sum_{j=1}^{J} f_{j,n}(\mathbf{x}_k) \right] .
\end{aligned} \tag{5.17}$$

Note that, $\forall k > n | T_k = T_n, S(\mathbf{x}, T_n) = S(\mathbf{x}, T_k) \Rightarrow s_k(\mathbf{x}) = 1$. □

Lemmas 3 and 4 extend steps 1 and 2 of the Kaplan-Meier model derivation.

5.1.3 The coupled baseline hazard model

In this section we introduce the coupled baseline hazard model. We derive a profile likelihood for the model and prove that maximizing it is equivalent to maximizing the likelihood within the class of all possible survival functions with the same input covariates.

Definition 2. *Given a survival dataset* $\mathcal{D} = \{\mathbf{x}_n; T_n; j_n; E_n | n \in \{1, 2, \ldots, N\}; T_n \leq T_m, \forall n < m; E_m = 0, \forall m > n, T_m = T_n\}$*, a coupled baseline hazard model for* $\mathcal{D}$ *is a piecewise constant survival model that only changes at the time points* T_n *and is specified by the functions:*

$$f_{j,n}(\mathbf{x}) = \begin{cases} 0 & , E_n = 0 \text{ or } j_n \neq j\,, \\ 1 - \left(\frac{\sum_{m=n+1}^{N} \omega_{j,n}(\mathbf{x}_m)}{\sum_{m=n}^{N} \omega_{j,n}(\mathbf{x}_m)} \right)^{\frac{\omega_{j,n}(\mathbf{x})}{\omega_{j,n}(\mathbf{x}_n)}} & , E_n = 1, j_n = j\,. \end{cases} \tag{5.18}$$

where $\omega_{j,n}(\mathbf{x})$ *is a strictly positive function of* $\mathbf{x}$ *for any* $n \in \{1, \ldots, N\}$, $j \in \{1, \ldots, J\}$.

The coupled baseline hazard model can be interpreted as a time-dependent and competing risks extension of the model proposed by [11] (equation 4.25 p. 86), and further developed by [84]. Indeed, if the log-linear hazard ratio is replaced by a generic function $\omega(\mathbf{x}, t)$, the Kalbfleish & Prentice model results in the coupled baseline hazard model for single risk scenarios. The demonstration in [11] assumes a time-invariant hazard ratio and a proportional hazard structure. Nonetheless, we show that this time-dependent extension can be obtained without any parametric assumption. In Theorem 2, we derive the profile likelihood for the coupled baseline hazard model and show that the model within this class that maximizes this likelihood also maximizes the likelihood among all survival models.

Theorem 2. *Given a survival dataset* $\mathcal{D} = \{\mathbf{x}_n; T_n; j_n; E_n | n \in \{1, 2, \ldots, N\}; T_n \leq T_m, \forall n < m; E_m = 0, \forall m > n, T_m = T_n\}$*, with random censoring, the likelihood of a coupled baseline hazard model is equal to the profile likelihood:*

$$\mathcal{L} = \prod_{n \in \{1,\ldots,N | E_i = 1\}} \Phi_{j_n,n} \left(1 - \Phi_{j_n,n}\right)^{\left(\Phi_{j_n,n}^{-1} - 1\right)}\,, \tag{5.19}$$

where:

$$\Phi_{j,n} = \frac{\omega_{j,n}(\mathbf{x}_n)}{\sum_{m=n}^{N} \omega_{j,n}(\mathbf{x}_m)}\,. \tag{5.20}$$

The supremum of the profile likelihood within the class of all coupled baseline hazard models is equal to the supremum of the likelihood in the class of all possible survival models in the form $S(\boldsymbol{x}, t)$.

Proof. The first claim in the theorem can be proven as follows. A coupled baseline hazard model follows the form given by equation (5.18):

$$f_{j,n}(\mathbf{x}) = 1 - \left(\frac{\sum_{k=n+1}^{N} \omega_{j,n}(\mathbf{x}_k)}{\sum_{k=n}^{N} \omega_{j,n}(\mathbf{x}_k)} \right)^{\frac{E_n 1(j_n=j)\omega_{j,n}(\mathbf{x})}{\omega_{j,n}(\mathbf{x}_n)}} = 1 - (1 - \Phi_{j,n})^{\frac{E_n 1(j_n=j)\omega_{j,n}(\mathbf{x})}{\omega_{j,n}(\mathbf{x}_n)}}. \quad (5.21)$$

Lemma 4 shows that the likelihood is given by:

$$\begin{aligned} \mathcal{L}\{\mathcal{D}|\mathbf{S}\} &= \prod_{n=1}^{N} \left[\left[1 - \sum_{j=1}^{J} f_{j,\nu[n]}(\mathbf{x}_n) \right]^{(1-E_n)} f_{j_n,\nu[n]}(\mathbf{x}_n)^{E_n} \prod_{k=\nu[n]+1}^{N} \left[1 - \sum_{j=1}^{J} f_{j,n}(\mathbf{x}_k) \right] \right] \\ &= \prod_{n \in \{1,\ldots,N | E_n=1\}} \left[f_{j_n,n}(\mathbf{x}_n) \prod_{k=n+1}^{N} \left[1 - f_{j,n}(\mathbf{x}_k) \right] \right] \\ &= \prod_{n \in \{1,\ldots,N | E_n=1\}} \Phi_{j_n,n} \left(1 - \Phi_{j_n,n} \right)^{(\Phi_{j_n,n}^{-1} - 1)}. \end{aligned} \quad (5.22)$$

The second claim can be proven as follows. Lemma 1 guarantees that, the supremum of the likelihood among the class $\mathcal{M}$ of all models that follow equation (1) is equal to the supremum of the likelihood among the class of all possible survival models. First, the supremum necessarily exists since the likelihood is by definition superiorly bounded by 1. Second, if $\mathcal{L}^*$ is the supremum of the likelihood among the class of all survival models, there is a sequence of models $\mathbf{S}^{(k)}$ so that $\lim_{k\to\infty} \mathcal{L}(\mathcal{D}|\mathbf{S}^{(k)}) = \mathcal{L}^*$. Using lemma 1, we construct a sequence $\mathbf{S}^{*(k)}$ so that $\mathcal{L}(\mathcal{D}|\mathbf{S}^{*(k)}) \geq \mathcal{L}(\mathcal{D}|\mathbf{S}^{(k)}) \Rightarrow \lim_{k\to\infty} \mathcal{L}(\mathcal{D}|\mathbf{S}^{(k)}) \geq \lim_{k\to\infty} \mathcal{L}(\mathcal{D}|\mathbf{S}^{(k)}) = \mathcal{L}^*$. Since $\mathcal{L}^*$ is the supremum likelihood among the class of all models, we have: $\Rightarrow \lim_{k\to\infty} \mathcal{L}(\mathcal{D}|S^{(k)}) \leq \mathcal{L}^*$. Hence, the supremum among $\mathcal{M}$ is also $\mathcal{L}^*$.

Writing this model in the form proposed in lemma 2, we define the class of models $\mathcal{M}^+$ so that $s_n(\mathbf{x}) > 0, \forall n, \mathbf{x}$ and either $f_{j,n}(\mathbf{x}) = 0, \forall \mathbf{x}$ or $f_{j,n}(\mathbf{x}) > 0, \forall \mathbf{x}$. The supremum of $\mathcal{M}^+$ is also $\mathcal{L}^*$, since it is possible to construct a model within $\mathcal{M}^+$ that has likelihood arbitrarily close to the likelihood of any model in $\mathcal{M}$. Within $\mathcal{M}^+$, it is possible to rewrite $f_{j,n}(\mathbf{x})$ in the form $f_{j,i}(\mathbf{x}) = \alpha_{j,n}^{\omega_{j,n}(\mathbf{x})}$, where $0 < \alpha_{j,n} \leq 1$, and $\omega_{j,n}(\mathbf{x}) > 0, \forall \mathbf{x}$. According to lemma 2, the likelihood of a model in $\mathcal{M}^+$ is

given by $\mathcal{L}\{\mathcal{D}|\mathbf{S}\} = \prod_{n=1}^{N} \mathcal{L}_n$, where:

$$\mathcal{L}_n = \begin{cases} \prod_{k=n}^{N} \left[1 - \sum_{j=1}^{J} \left[1 - \alpha_{j,\nu[n]}^{\omega_{j,\nu[n]}(\mathbf{x}_n)}\right]\right] & , E_n = 0\,, \\ \left[1 - \alpha_{j_n,n}^{\omega_{j_n,n}(\mathbf{x}_n)}\right] \prod_{k=\nu[n]+1}^{N} \left[1 - \sum_{j=n}^{J} \left[1 - \alpha_{j,\nu[n]}^{\omega_{j,\nu[n]}(\mathbf{x}_n)}\right]\right] & , E_n = 1\,. \end{cases} \tag{5.23}$$

If $E_n = 0$ and $E_k = 0, \forall k | T_n = T_k$, we have: $f_{j,k}(\mathbf{x}) = 0, \forall j,k | k > n; T_k = T_n \Rightarrow \alpha_{j,k} = 1, \forall j,k | k > n; T_k = T_n$. Thus $\mathcal{L}_{T_n}$ is maximized by making $p_{j,n} = 1$.

If $E_n = 1$: $T_k < T_n | k < n$ and there might be values of k so that $T_k = T_n$, $k > n$ and $E_k = 0$; $f_{j,k}(\mathbf{x}) = 0, \forall j,k | k > n; T_k = T_n \Rightarrow \alpha_{j,k} = 1, \forall j,k | k > n; T_k = T_n$. Thus $\mathcal{L}_{T_n}$ is maximized by making $\alpha_{j_n,n} = \Phi_{j_n,n}^{1/\omega_{j_n,n}(\mathbf{x}_n)}$ and $\alpha_{j,n} = 1 \forall j \neq j_n$.

Finally, this means that, for any survival model $\mathbf{S}$ in $\mathcal{M}^+$, it is possible to build a coupled baseline hazard model $\mathbf{S}^c$ by making:

- $\omega_{j,n}^c(\mathbf{x}) = \omega_{j,n}(\mathbf{x})$;
- $\alpha_{j_n,n}^c = \Phi_{j_n,n}^{1/\omega_{j_n,n}(\mathbf{x}_n)}, \forall n | E_n = 1$;
- $\alpha_{j,k}^c = 1, \forall j,n | (j \neq j_n) or (E_n = 0)$.

which will ensure that $\mathcal{L}\{\mathcal{D}|\mathbf{S}^c\} \geq \mathcal{L}\{\mathcal{D}|\mathbf{S}\}$. Once again, it is possible to show that the supremum of the likelihoods of models in $\mathcal{M}^+$ is equal to the supremum of the likelihoods coupled baseline hazard models, since given a sequence of models $\mathbf{S}^{(n)} \in \mathcal{M}^+$ that converges to the supremum $\mathcal{L}^*$ it is possible to build a sequence of coupled baseline hazard models $\mathbf{S}^{c(n)}$ that also converges to $\mathcal{L}^*$. Hence, the supremum of the likelihoods of coupled baseline hazard models is equal to the supremum is likelihoods of models in the class of all possible survival models. □

5.1.4 Cause specific hazard and cumulative incidence

Although the coupled baseline hazard model has been derived using an unconventional notation chosen to facilitate theoretical derivation, it is possible to convert it to other notations. The cumulative incidence function for a coupled

baseline hazard model can be retrieved as follows:

$$\begin{aligned} F_j(\mathbf{x},t) =& Pr\{T \leq t, j|\mathbf{x}\} = \sum_{n|T_n \leq t} [S(\mathbf{x}, T_{v[n]-1}) - S(\mathbf{x}, T_n)] p_j(\mathbf{x}, T_n) \\ =& \sum_{n|T_n \leq t} f_{j,n}(\mathbf{x}) S(\mathbf{x}, T_{v[n]-1}) \\ =& \sum_{n|T_n \leq t, E_n=1, j_n=j} \left[1 - \left(\frac{\sum_{m=n+1}^{N} \omega_{j,n}(\mathbf{x}_m)}{\sum_{m=n}^{N} \omega_{j,n}(\mathbf{x}_m)} \right)^{\frac{\omega_{j,n}(\mathbf{x})}{\omega_{j,n}(\mathbf{x}_n)}} \right] S(\mathbf{x}, T_{n-1}), \end{aligned} \tag{5.24}$$

where:

$$S(\mathbf{x}, T_{n-1}) = \prod_{m|m<n, E_m=1} \left(\frac{\sum_{k=m+1}^{N} \omega_{j_m,m}(\mathbf{x}_k)}{\sum_{k=m}^{N} \omega_{j_m,m}(\mathbf{x}_k)} \right)^{\frac{\omega_{j_m,m}(\mathbf{x})}{\omega_{j_m,m}(\mathbf{x}_m)}}. \tag{5.25}$$

The cause specific hazard for a coupled baseline hazard model can be expressed in terms of the Dirac delta function, since the survival function is not continuous. Alternatively, the cause specific survival function can be expressed as follows:

$$\begin{aligned} S_j(\mathbf{x},t) =& Pr\{T > t|\mathbf{x}, j\} = \exp\left(\int_0^t \lambda(\mathbf{x}, u) \mathrm{d}u \right) = \prod_{n \in \{1,\ldots,N\}|T_n \leq t} 1 - f_{j,n}(\mathbf{x}) \\ =& \prod_{n \in \{1,\ldots,N\}|T_n \leq t, E_n=1, j_n=j} \left(\frac{\sum_{m=n+1}^{N} \omega_{j,n}(\mathbf{x}_m)}{\sum_{m=n}^{N} \omega_{j,n}(\mathbf{x}_m)} \right)^{\frac{\omega_{j,n}(\mathbf{x})}{\omega_{j,n}(\mathbf{x}_n)}}. \end{aligned} \tag{5.26}$$

5.1.5 The PH-MNN model as an specific implementation of the coupled baseline hazard model

In principle, it would be possible to perform nonparametric maximum likelihood estimation of the functions $\omega_{j,n}(\mathbf{x})$ while imposing no restriction on their covariate dependence. In this case, the likelihood would be maximized by making $\omega_{j,n}(\mathbf{x}) \to 0$ for any $\mathbf{x} \neq \mathbf{x}_n$ and any $j \neq j_n$. The outcome of this procedure would be to create a stratified Kaplan-Meier estimator in which a different estimator is generated for every possible value of $\mathbf{x}$. This is not a viable solution in most applications, as there are insufficient subjects with identical values for all covariates. This creates the need of a parametric representation of $\omega_{j,n}(\mathbf{x})$. For that, we first replace the time index by a continuous time variable in the form $\omega_j(\mathbf{x}, t)$. There, the original formulation can be retrieved by making $\omega_{j,n}(\mathbf{x}) = \omega_j(\mathbf{x}, T_n)$.

Note that this change does not affect the original model because, although ω_j was originally only defined for time points in which an event has been observed within the dataset, this does not create any constraints for the function in the points that are used in the model. With this continuous time representation, a semiparametric implementation of the coupled baseline hazard function can be obtained by making a parametric representation of $\omega_j(\mathbf{x}, t)$. Nevertheless, computation in the model for a single input variable $\mathbf{x}$ requires the function $\omega_j(\mathbf{x}, t)$ to be computed for a set of t that grows linearly with the number of subjects in the dataset. If a standard neural network was used to represent it, the computation would easily become unfeasible. Instead, we use a metaparametric neural network that has t as an explicit variable, leading to the PH-MNN model. There, a basis function representation can be used to represent the time dependency so that re-computation of the neural network is not necessary as detailed in [70]. Given Theorem 1, we see that if the number of parameters in the model was infinite, the PH-MNN model would approach a nonparametric model. Therefore, any limitation in the capability of the PH-MNN to represent a target probability distribution is caused by the limited number of parameters adopted in a particular instance of it and not by a limitation in the structure of the model itself.

5.1.6 Relationship with other survival models

The coupled baseline model was derived as a means of justifying the structure of the PH-MNN model. Nonetheless, other model also be retrieved as less generic versions of it. This shows that the coupled baseline hazard model is a generic framework for survival analysis that unifies the theoretical background of various models.

Relationship with the Kaplan-Meier model

The derivation for the maximum likelihood property of the coupled baseline hazard model is an extension of the maximum likelihood derivation of the Kaplan-Meier model provided by [6]. As a result, the Kaplan-Meier model can be retrieved if it is assumed that the survival function does not depend on $\mathbf{x}$. Indeed, if in a single risk model $\omega_{1,n}(\mathbf{x}) = w_{1,n} > 0, \forall n \in \{1, \ldots, N\}$, where $w_{1,n}$ is a positive

constant, equation (5.18) is reduced to the Kaplan-Meier estimator:

$$S(\mathbf{x}, t) = \prod_{n|T_n \leq t} \left(\frac{N-n}{N-n+1} \right)^{E_n} . \tag{5.27}$$

In the case of a competing risks scenario, the Kaplan-Meier model also provides the maximum likelihood estimator for the survival probability distribution, as discussed in [7]. This estimator can be retrieved from the coupled baseline hazard model by making $\omega_{j,n}(\mathbf{x}) = w_{j,n} > 0, \forall n \in \{1, \ldots, N\}, j \in \{1, \ldots, J\}$, where $w_{j,n}$ are positive constants.

Relationship with the Cox proportional hazards model

The Cox proportional hazards model can also be retrieved as a special case of the coupled baseline hazard model. In a single risk scenario in which $\omega_{1,n}(\mathbf{x}) = \exp(\boldsymbol{\beta}^T \mathbf{x}), \forall n \in \{1, \ldots, N\}$, the coupled baseline hazard model can be expressed as $\Lambda(\mathbf{x}, t) = \Lambda_0(t) \exp(\boldsymbol{\beta}^T \mathbf{x})$, where $\Lambda_0(t)$ is the nonparametric estimator for the baseline hazard function defined in [11]. Additionally, the partial likelihood can be related to the log profile likelihood as follows:

$$\log(\mathcal{L}) = \sum_{n \in \{1,\ldots,N|E_n=1\}} \log(\Phi_n) + \left(\Phi_n^{-1} - 1 \right) \log(1 - \Phi_n) , \tag{5.28}$$

where the log partial likelihood is:

$$\log(\Phi_n) = \boldsymbol{\beta}^T \mathbf{x}_n - \log \left(\sum_{k=n}^{N} \exp(\boldsymbol{\beta}^T \mathbf{x}_k) \right) . \tag{5.29}$$

In Section 5.2.1 we show that both estimators have equivalent large sample properties and propose a new objective function that also has equivalent large sample properties, but with reduced small sample bias.

Relationship with other competing risks models

In a competing risks scenario the proportional cause specific hazard model [3] can be obtained from an approximation of the coupled baseline hazard model using an analogous approach. On the other hand, a proportional subdistribution hazard model [4] cannot be obtained from the coupled baseline hazard

model. This is because the likelihood factorization used to prove Theorem 2 is not possible within a proportional subdistribution hazard framework.

The representation used for the survival function in the proof of Theorem 2 has a similar form to a mixture model. Nevertheless, this model cannot be retrieved directly from the coupled baseline hazard model. Alternatively, it is possible to adapt Theorem 2 towards achieving a model that has a closer representation to a mixture model. In this case, the resulting model can be represented by the functions:

$$f_{j,n}(\mathbf{x}) = \begin{cases} 0 & , E_n = 0\,, \\ p_{j,n}(\mathbf{x}) \left[1 - \left(\frac{\sum_{m=n+1}^{N} \omega_n(\mathbf{x}_m)}{\sum_{m=n}^{N} \omega_n(\mathbf{x}_m)} \right)^{\frac{\omega_n(\mathbf{x})}{\omega_n(\mathbf{x}_n)}} \right] & , E_n = 1\,, \end{cases} \tag{5.30}$$

and the profile likelihood will be given by:

$$\mathcal{L} = \prod_{n \in \{1,\ldots,N | E_n = 1\}} p_{j,n}(\mathbf{x}) \Phi_n \left(1 - \Phi_n\right)^{\left(\Phi_n^{-1} - 1\right)}\,, \tag{5.31}$$

where $\Phi_n = \omega_n(\mathbf{x}_n) / \sum_{m=n}^{N} \omega_n(\mathbf{x}_m)$.

The main disadvantage of this alternative form is that the distributions for all risks are based on the same baseline cumulative hazard function. If the shape of the distribution for each risk is different, this has to be compensated by the functions $p_{j,n}(\mathbf{x})$. In particular, if there is a relevant high frequency difference between the distributions for each risk, the functions $p_{j,n}(\mathbf{x})$ will require high frequency components to compensate for it.

5.2 Asymptotic properties

5.2.1 Asymptotic equivalence theorem

Given that the profile likelihood proposed here is analogous to the Cox partial likelihood, in this section we investigate the relationship between the asymptotic properties of both estimators. For that purpose, we define a class of transformations that can be performed on the partial likelihood without changing the asymptotic behavior of the estimator. For simplicity, we restrict the study of this transformation to the specific case in which the neural network block in the PH-

MNN model is replaced by a linear model. Similarly to other neural network models, studying the asymptotic behavior of a PH-MNN model is challenging because there is ambiguity in the combination of weights that can lead to the same output function. Nonetheless, the study of the linear version of it covers the specific difference between MNNs and standard neural networks showing that if the feature extraction is fixed, the top layer of the PH-MNN models would converge.

Theorem 3. *Given a proportional subdistribution hazard model in the form:*

$$\lambda_j(t, \mathbf{x}_n) = \lambda_j(t, [x_{1,n}, \ldots, x_{I,n}]) = \lambda_{j,0}(t) \exp\left(\sum_{i=1}^{I} \sum_{k=1}^{K} \beta_{i,j,k} g_k(t) x_{n,i} \right), \tag{5.32}$$

where:

1. $|g_k(t)| \leq G, \forall t, k.$
2. $\exists C, t > 0 \mid \forall Y > 0,\ Pr[\exp(x_{n,i}) > Y] \leq C\exp(-tY),\ Pr[\exp(-x_{n,i}) > Y] \leq C\exp(-tY).$
3. $\log(\Phi) = \sum_{n=1}^{N} E_n \log(\Phi_n)$, *where* $\log(\Phi)$ *is the log partial likelihood,* E_n *is the event indicator,* $\log(\Phi_n) = \omega_{j_n}(\mathbf{x}_n, T_n) / \sum_{m=n}^{N} \omega_{j_n}(\mathbf{x}_m, T_n)$ *and* $\omega_j(\mathbf{x}_n, t) = \exp(\sum_{i=1}^{I} \sum_{k=1}^{K} \beta_{i,j,k} g_k(t) x_{n,i})$.
4. *two different subjects do not experience an event at the exact same time.*
5. *constants* $p \geq 1$ *and* $q_{i,j,k} > 0$.

Let $h(u)$ *be a function defined in* $(-\infty, 0]$ *which is differentiable in* $(-\infty, 0)$ *and has its derivative bounded by* $|1 - h'(u)| \leq \alpha/(\exp(-u) - 1)$, $\alpha > 0$, $\forall u < 0$.

The function $N^{-1/2}\left[\nabla\left(\sum_{i,j,k} q_{i,j,k}|\beta_{i,j,k}|^p + \sum_{n=1}^{N-1} E_n h(\log(\Phi_n))\right) - \nabla \log(\Phi)\right]$ *will converge in probability to* 0 *for* $N \to \infty$ *in any bounded subspace of* $\mathbf{fi} = [\beta_{i,j,k} | i \in \{1, \ldots, I\}; j \in \{1, \ldots, J\}; k \in \{1, \ldots, K\}]$.

Proof. The absolute value of the difference between both score functions is

bounded by:

$$\left|\frac{\partial}{\partial\beta_{i,j,k}}\left[-\sum_{i,j,k}q_{i,j,k}|\beta_{i,j,k}|^p+\sum_{n=1}^{N-1}\left(\log(\Phi_n)-h(\log(\Phi_n))\right)\right]\right|\leq$$
$$q_{i,j,k}B^{p-1}p+\sum_{n=1}^{N-1}\left|(1-h'(\log(\Phi_n)))\frac{\partial}{\partial\beta_{i,j,k}}\log(\Phi_n)\right|\leq$$
$$q_{i,j,k}B^{p-1}p+\sum_{n=1}^{N-1}\frac{E_n\alpha}{\Phi_n^{-1}-1}\left|\frac{\sum_{m=n}^{N}g_k(T_n)(x_{n,i}-x_{m,i})\omega_{j(i)}(\mathbf{x}_m,T_n)}{\sum_{m=n}^{N}\omega_{j(i)}(\mathbf{x}_m,T_n)}\right|=$$
$$q_{i,j,k}B^{p-1}p+\sum_{n=1}^{N-1}\frac{E_n\alpha\omega_{j_n}(\mathbf{x}_n,T_n)|g_k(T_n)|}{\sum_{m=n+1}^{N}\omega_{j_n}(\mathbf{x}_m,T_n)}\left|\frac{\sum_{m=n}^{N}(x_{n,i}-x_{m,i})\omega_{j(i)}(\mathbf{x}_m,T_n)}{\sum_{m=n}^{N}\omega_{j_n}(\mathbf{x}_m,T_n)}\right|\leq$$
$$q_{i,j,k}B^{p-1}p+\sum_{n=1}^{N-1}\frac{\alpha\omega_{j_n}(\mathbf{x}_n,T_n)GX}{\sum_{m=n+1}^{N}\omega_{j(i)}(\mathbf{x}_m,T_n)}\leq$$
$$q_{i,j,k}B^{p-1}p+\alpha GX\frac{\max_n\omega_{j_n}(\mathbf{x}_n,T_n)}{\min_m\omega_{j_n}(\mathbf{x}_m,T_n)}\sum_{n=1}^{N-1}\frac{1}{N-n}\leq$$
$$q_{i,j,k}B^{p-1}p+\alpha\exp\left(IKBGX\right)GX\sum_{n=1}^{N-1}\frac{1}{N-n}\leq$$
$$q_{i,j,k}B^{p-1}p+\alpha\exp\left((IKB+1)GX\right)\log(N)\,, \tag{5.33}$$

where I is the length of vector $\mathbf{x}_n$; K is the number of functions $g_k(t)$; B is an upper bound for $\beta_{i,j,k}$; and $X=\max_{n,m,i}|x_{n,i}-x_{m,i}|$.

For each component i of $\mathbf{x}$, the probability distribution of X can be bounded as follows:

$$Pr[\exp(\max_{n,m,i}(x_{n,i}-x_{m,i}))>Y]\leq$$
$$Pr[\exp(\max_{n,i}(x_{n,i}))>Y/2]+Pr[\exp(-\min_{n,i}(x_{n,i}))>Y/2]\leq$$
$$NI(Pr[\exp(x_{n,i})>Y/2]+Pr[\exp(-x_{n,i})>Y/2])\leq$$
$$2NIC\exp(-tY/2)\,. \tag{5.34}$$

Therefore, for any $\epsilon > 0$:

$$
\begin{aligned}
&Pr\left[\left|N^{-1/2}\frac{\partial}{\partial\beta_{i,j,k}}\left[-\sum_{i,j,k}q_{i,j,k}|\beta_{i,j,k}|^p+\sum_{n=1}^{N-1}\left(\log(\Phi_n)-h(\log(\Phi_n))\right)\right]\right|>\epsilon\right]\leq\\
&Pr\left[N^{-1/2}\left[q_{i,j,k}B^{p-1}p+\alpha\exp\left((IKB+1)GX\right)\log(N)\right]>\epsilon\right]=\\
&Pr\left[\exp(X)>\left(\frac{\epsilon\sqrt{N}-q_{i,j,k}B^{p-1}p}{\alpha\log(N)}\right)^{\frac{1}{(IKB+1)G}}\right]\leq\\
&2NIC\exp\left(-\frac{t}{2}\left[\left(\frac{\epsilon\sqrt{N}-q_{i,j,k}B^{p-1}p}{\alpha\log(N)}\right)^{\frac{1}{(IKB+1)G}}\right]\right)=\\
&2IC\exp\left(\left(1-\frac{\alpha t\left(\epsilon\sqrt{N}-q_{i,j,k}B^{p-1}p\right)^{\frac{1}{(IKB+1)G}}}{2\left(\alpha\log(N)\right)^{\frac{1}{(IKB+1)G}+1}}\right)\log(N)\right).
\end{aligned}
\tag{5.35}
$$

Since $\lim_{N\to\infty}N^a/\log^b(N)=\infty,\forall a,b>0$, we have for all values of i, j and k that:

$$
\lim_{N\to\infty}Pr\left(\left|N^{-1/2}\frac{\partial}{\partial\beta_{i,j,k}}\left(-\sum_{i,j,k}q_{i,j,k}|\beta_{i,j,k}|^p+\sum_{n=1}^{N-1}\left(\log(\Phi_n)-h(\log(\Phi_n))\right)\right)\right|>\epsilon\right]=0\,. \tag{5.36}
$$

Therefore

$$
\lim_{N\to\infty}Pr\left[\left|N^{-1/2}\left(\nabla\log(\Phi)-\nabla\left(\sum_{i,j,k}q_{i,j,k}|\beta_{i,j,k}|^p+\sum_{n=1}^{N-1}E_nh(\log(\Phi_n))\right)\right)\right|>\epsilon\right]=0\,. \tag{5.37}
$$

□

As a consequence the proposed profile likelihood will have equivalent large sample behavior to the partial likelihood, given that the probability distribution of x follows the appropriate condition. For example, a Gaussian distribution satisfies this condition, since for $x\sim\mathcal{N}(\mu,\sigma^2)$ neither $\exp(x)$ or $\exp(-x)$ will have a heavy-tailed distribution. Also, the asymptotic properties will be preserved with both nonlinear transformations that follow the conditions in the theorem and with arbitrary regularizers from L^1 to L^∞.

5.2.2 Sample complexity of PH-MNN model estimation

In this section we present a brief discussion of the the sample complexity of estimating the PH-MNN model. Based on Theorem 3, we have restricted this analysis for the case of partial likelihood estimation. The sample complexity can be defined as the amount of data necessary to achieve a given maximum error ϵ in estimation with probability $1-\delta$ [87]. Similarly to Theorem 3, we restrict the study to the outer layer of the PH-MNN model. There, the target is to estimate the hazard ratio as a function of time and covariates.

From Theorem 8.4.4 of [88] we have the conditions under which it is possible to guarantee the asymptotic distribution of $\beta_{i,j,k}$. Despite basis functions not being used there, the same results can be achieved by defining:

$$z_{n,i,k}(t) = g_k(t)x_{n,i}\,. \tag{5.38}$$

Then, $z_{n,i,k}(t)$ will be equivalent to time-dependent covariates and the asymptotic distribution of $\boldsymbol{\beta}$ will be:

$$n^{1/2}(\boldsymbol{\beta}-\boldsymbol{\beta}_0) \to \mathcal{N}(0, \Sigma^{-1}(\boldsymbol{\beta}_0, \tau))\,, \tag{5.39}$$

as $n \to \infty$, where τ is the follow up time horizon in the model.

The error in the log-hazard ratio for risk j is given by:

$$E_j = \|\log\omega_j(\mathbf{x},t) - \log\omega_{j,0}(\mathbf{x},t)\| = \|(\boldsymbol{\beta}_j - \boldsymbol{\beta}_{j,0})\mathbf{x}\|\,, \tag{5.40}$$

where $\omega_{j,0}(\mathbf{x},t)$ indicates the true hazard ratio for event j and $\boldsymbol{\beta}_{j,0}$ indicates the true values of $\boldsymbol{\beta}_j$. Thus, the error is bounded by:

$$E_j \leq \sum_{i,k} \|\beta_{i,j,k} - \beta_{i,j,k,0}\| GX\,, \tag{5.41}$$

Where X is the upper boundary for x_i and G is the upper boundary for $g_k(t)$. For a target error boundary ϵ, we assume that all dimensions of $\boldsymbol{\beta}_j$ must have error less or equal to $\epsilon/(IKGX)$, where I is the dimension of $\boldsymbol{x}$ and K is the number of basis functions $g_k(t)$. For each dimension, the probability of the error being

larger than the threshold will be bounded by:

$$\lim_{n\to\infty} P\left(|\beta_{i,j,k} - \beta_{i,j,k,0}| \geq \frac{\epsilon}{IGKX}\right) \leq \operatorname{erfc}\left(\frac{\epsilon\sqrt{n}}{IGKX\sigma_{\max}(\boldsymbol{\beta}_0,\tau)\sqrt{2}}\right), \tag{5.42}$$

where $\sigma_{\max}(\boldsymbol{\beta}_0,\tau)$ is the standard deviation in the principle direction of $\mathcal{N}(0,\Sigma^{-1}(\boldsymbol{\beta}_0,\tau))$. Then, for E_j to be bounded by ϵ with probability $1-\delta$ in the limit for $n\to\infty$, the following condition suffices:

$$\operatorname{erfc}\left(\frac{\epsilon\sqrt{n}}{IGKX\sigma_{\max}(\boldsymbol{\beta}_0,\tau)\sqrt{2}}\right) \leq \frac{\delta}{IK}, \tag{5.43}$$

which is equivalent to:

$$n \geq 2\left(\frac{IGKX\sigma_{\max}(\boldsymbol{\beta}_0,\tau)\operatorname{erfc}^{-1}(\delta/IK)}{\epsilon}\right)^2. \tag{5.44}$$

Thus, the asymptotic limit of the sample complexity of the PH-MNN model (outer layer) is superiorly bounded by $\mathcal{O}((\operatorname{erfc}^{-1}(\delta/IK)/\epsilon)^2)$.

5.2.3 Small sample bias minimization

Based on the existence of a family of estimators with equivalent asymptotic behaviour, we now investigate which estimator leads to the smallest small sample bias. Although no described method provides an analytical expression for the best possible estimator, an improvement over the partial likelihood estimator can be achieved through the following procedure.

We define the log reduced likelihood as $r = \sum_{n=1}^{N-1} E_n r_n$, where E_n is the indicator of censoring and:

$$r_n = \sum_{k,i} \beta_{i,j_n,k} f_k(T_n) x_{n,i} - \log\left(\sum_{m=n+1}^{N-1} \exp\left(\sum_{k,i} \beta_{i,j_n,k} f_k(T_n) x_{m,i}\right)\right). \tag{5.45}$$

Each component of the partial likelihood is obtained from the respective component of the reduced likelihood, through the expression $\log(\Phi_n) = -\log(1+\exp(-r_n))$. Although the large sample behavior for both the reduced and the partial likelihood estimators is equivalent, as shown in Theorem 3, the definition of the reduced likelihood clarifies the small sample behavior of the partial likeli-

hood. In particular, we study the behavior of the reduced likelihood for extreme values of some of the parameters $\beta_{i,j_n,k}$. If the other parameters are bounded:

$$\lim_{\beta_{i,j_n,k}\to\infty}\left[r_n-\beta_{i,j_n,k}\left(f_k(T_n)x_{n,i}-\max_{m>n}\left[f_k(T_n)x_{m,i}\right]\right)\right]=0\,, \tag{5.46}$$

and:

$$\lim_{\beta_{i,j_n,k}\to-\infty}\left[r_n-\beta_{i,j_n,k}\left(f_k(T_n)x_{n,i}-\min_{m>n}\left[f_k(T_n)x_{m,i}\right]\right)\right]=0\,. \tag{5.47}$$

The behavior of the likelihood for extreme values of $\beta_{i,j_n,k}$ is illustrated in Fig. 5.1. The probability distribution of $f_k(T_n)x_{n,i}$ for subject n will be different from the probability distribution of other subjects at risk at T_n. This is because the event probability is proportional to $f_k(T_n)x_{n,i}$. Consequently, the smaller a dataset is the more likely it is for $f_k(T_n)x_{n,i}$ to fall outside the closed interval from $\min_{m>n}\left[f_k(T_n)x_{m,i}\right]$ to $\max_{m>n}\left[f_k(T_n)x_{m,i}\right]$ (Figure 5.1(a)). If this is the case, the likelihood diverges to ∞ for either $\beta_{i,j_n,k}\to\infty$ or $\beta_{i,j_n,k}\to-\infty$, making the estimation of $\beta_{i,j_n,k}$ diverge. On the other hand, the larger a dataset is the less likely this is to happen, making the reduced likelihood converge to 0 and preventing $\beta_{i,j_n,k}$ from divergence (Fig. 5.1(b)).

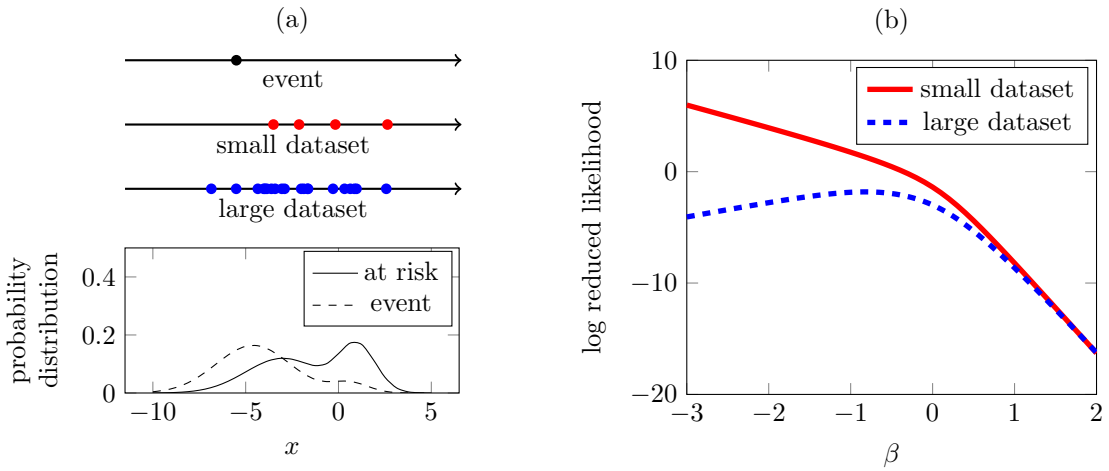


Figure 5.1.: Illustration of the effect of the dataset size on the log reduced likelihood function. For simplicity, a scenario with unidimensional covariate and time-constant hazard ratio is assumed. In panel (a), the horizontal axis of all four graphs represents the values of the covariate x. In the uppermost axis, the dot represents x for a subject that has experienced the event at time T_n, whilst in the second and third axis, the dots represent x for subjects at risk immediately after time T_n in a small or large dataset respectively. The bottom graph shows the probability distributions of x for subject who experienced the event at time T_n (dashed line), and for other subjects at risk immediately after T_n (solid line). Panel (b) illustrates how the size of the dataset in panel (a) reflect on the n^{th} component of the log reduced likelihood.

This issue is partly solved in the partial likelihood by the transformation of the log

reduced likelihood r_n through the function $\log(\Phi_n) = -\log(1+\exp(-r_n))$. This transformation limits the components of the log likelihood to a limiting value of 0. This causes a bias because it is still possible that the log likelihood will be monotonic for the entire range of a parameter $\beta_{i,j_n,k}$, but with the advantage that it is less likely to happen. Indeed, one single sample n whose likelihood converges to 0 for both extremes of $\beta_{i,j_n,k}$ will be enough to prevent the estimation from diverging. A similar scenario appears with the profile likelihood used in this work, since the profile likelihood can be obtained from the reduced likelihood through the transformation $\log(\mathcal{L}_n) = -\log(1+\exp(-r_n)) - \exp(-r_n)\log(1+\exp(r_n))$. This will induce equivalent asymptotic behavior to $\log(\Phi_n) = -\log(1+\exp(-r_n))$.

This asymptotic analysis suggests the use of an improved likelihood that has the property of not only limiting the likelihood to 0, but also introduces a small slope with opposite sign whenever $r_n > 0$. The balanced likelihood achieves this by defining components B_n as:

$$\log(B_n) = -\log(1+\exp(-r_n)) - \alpha\log(1+\exp(r_n))\,, \tag{5.48}$$

where α is a positive hyper-parameter. When $\alpha = 0$, the balanced likelihood becomes equal to the partial likelihood. This objective function leads to a reduced bias and prevents the estimator from diverging, while being equivalent to both the partial and the profile likelihood estimators for large datasets as shown in Theorem 3. This estimator for the hazard ratio, when combined with the coupled baseline hazard model, results in the balanced coupled baseline hazard model.

5.3 Experimental investigation of the sample size effect

The asymptotic behavior of the PH-MNN model was evaluated using a synthetic competing risks dataset with nonlinear and time-dependent survival function. The pseudo-code for the PH-MNN model is provided in Algorithm 5 in Appendix A. The structure of the neural network block of the PH-MNN model is provided in Figure 4.1. Data was generated from a proportional subdistribution hazard model with nonlinear and time-dependent hazard ratio:

$$\tilde{\lambda}_1(t,\mathbf{x}) = 0.03(1+0.5\cos(2\pi t/10))\exp(\tan^{-1}(2x[0])\mathbb{1}(t<5) + \tan^{-1}(2x[1])\mathbb{1}(t>5))\,, \tag{5.49}$$

$$\tilde{\lambda}_2(t,\mathbf{x}) = 0.03(1+0.5\sin(2\pi t/10))\exp(\sin(x[1])\mathbb{1}(t<5)+\sin(x[0])\mathbb{1}(t>5))\,, \tag{5.50}$$

where $[x[0], x[1]] \sim \mathcal{N}(0, I)$. Censoring was included with uniform probability distribution for the censoring time within the interval $[0, 10]$.

Fig. 5.2 shows expected value of the mean integrated square error for the PH-MNN, nested PH-MNN and Cox models. For each model, three different versions were trained with different likelihoods: partial, profile and balanced. A sequence of progressively larger dataset sizes was generated, and 100 independent realizations were generated for each dataset size. Estimation was performed for each model with the same 100 random realizations of the dataset. In the nested PH-MNN model, the neural network component, $\psi(\mathbf{x};\theta)$ had $x[0]$ as its only input variable, with $\mathbf{y}_1$ being set to be $\mathbf{x}[1]$ and $\mathbf{y}_2$ the observation time. Figure 5.2 shows that a clear improvement is obtained with the nested structure. The likely reason of it the introduction of a bottleneck that makes it harder for the model to overfit in regions where little data is available. However, the different likelihoods didn't have a significant impact in training with small data sizes.

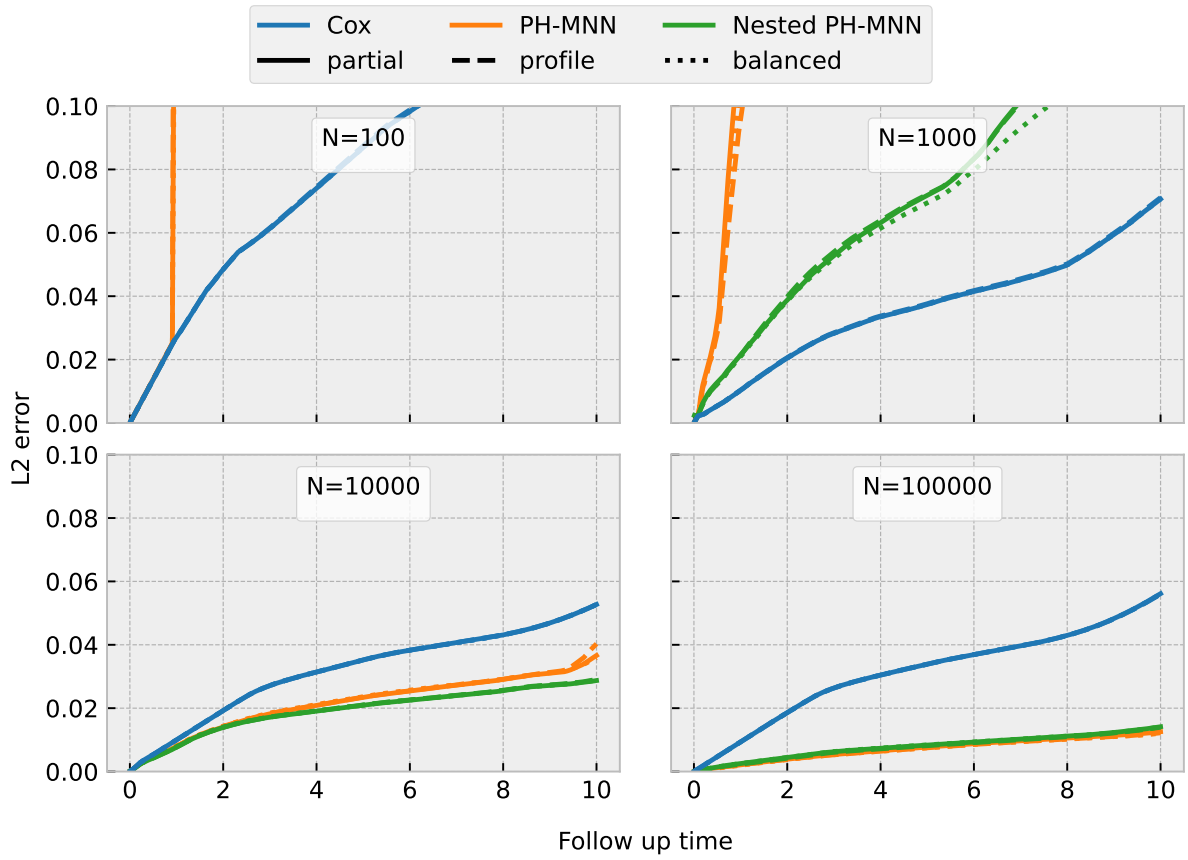


Figure 5.2.: L2 error of the cumulative incidence function as a function of the dataset size.

This suggests that the structure of the model might play a more important role in reducing training bias than the loss function.

6

Survival Modelling of Joint Replacement Surgeries

The main purpose of hip and knee replacement surgeries in the treatment of osteoarthritis is to reduce pain and improve functioning of the replaced joint, which should ultimately lead to an improvement to the patient's quality of life. Similarly to any surgical procedure, joint replacements are not free from risks so the decision of whether or not surgery is the best treatment should be based on a trade-off that takes into account the main risks and benefits of it. Con-

Table 6.1.: Number of complete data observations and events for the unlinked (without PROMs as input) and linked (with PROMs as input) hip replacement datasets. Revisions were only included when they have occurred within 10 years from the primary joint replacement.

Dataset	**Unlinked**	**Linked**
Number of patients	504415	206913
aseptic loosening/lysis	1537	677
dislocation/subluxation	1715	732
fracture	1228	530
infection	1604	684
pain	347	168
other	1057	431
deaths within 1 year	4792	1950

Table 6.2.: Number of complete data observations and events for the unlinked (without PROMs as input) and linked (with PROMs as input) knee replacement datasets. Revisions were only included when they have occurred within 10 years from the primary joint replacement.

Dataset	**Unlinked**	**Linked**
Number of patients	735071	255454
aseptic loosening/lysis	4006	1142
dislocation/subluxation	541	114
fracture	464	161
infection	3480	1270
instability	1921	684
pain	1803	530
progressive arthritis	2153	461
stiffness	529	228
other	2696	790
deaths within 1 year	6255	2214

sidering that this decision involves a balance between different aspects of a patient's life, the best decision will require an active role of the patient, who in most cases will not have in depth knowledge of the possible outcomes of the surgery. Therefore, conveying that knowledge to the patients is an important part of this decision process.

In the present chapter, we apply the nested PH-MNN model to the estimation of mortality and revision risks after hip and knee replacement surgeries in patients with osteoarthritis using data collected by the National Joint Registry (NJR). Additionally to the covariates present in the NJR dataset, we also use the preoperative patient reported outcome measures (PROMs) score as input for the survival models. PROMs are numeric evaluations of some aspect of a patient's health. This is obtained through a questionnaire that is answered directly by patients. The PROMs scores used in the model were the Oxford Hip Score (OHS) [89] and the Oxford Knee Score (OKS) [90] for hip and knee replacement surgeries respectively. The OHS a and OKS inputs were the answers to individual questions in the questionnaire instead of the overall scores. For the mortality model, we have performed single risk estimation with a maximum follow up of 1 year, since any deaths that happen after that period will most likely not be caused by the surgery. For the revision risk, we have divided the outcomes in different event types depending on the main indication for revision. The maximum follow up time for the revision models was 10 years. One important challenge

with the competing risks revision models is the small number of events observed in the complete data. In the present study, only complete data was used and data from joint replacements where not all input variables were known were not taken into account. Tables 6.1 and 6.2 show the complete dataset size for hip and knee replacements both when using or not preoperative PROMs as inputs. Then smaller dataset size when preoperative PROMs are used as inputs was overcame with a transfer learning strategy that was possible due to the nested PH-MNN model hierarchical structure. This strategy has enabled the use of data from procedures for which the preoperative PROMs is not available, minimizing the overfit.

6.1 Nested model structure

The present analysis was split into four models, one for each combination of joint type (hip or knee) and outcome of interest (mortality or revision). Although the PH-MNN model is a competing risks model and could in principle estimate mortality and revision risks simultaneously, it was preferred to split the model into two because the timescale of interest for the followup is different for each case. In the case of mortality risk, a follow up of 1 year was adopted since deaths directly related to orthopedic surgeries happen within the first year after the surgery. In the case of revision risk, the long term behavior of the implant is of great interest when choosing a treatment option so a follow up time of 10 years was adopted. The revision risk model was split into different indications for revision as competing risks allowing a more detailed analysis of the effect produced by any input variable of the model. The hip revision model, the competing risks were: aseptic loosening/lysis, dislocation/subluxation, fracture, infection, pain and other. For the knee revision model, the risks were: aseptic loosening/lysis, dislocation/subluxation, fracture, infection, instability, pain, progressive arthritis, stiffness and other. The complete list of input variables in the model are presented in Figures 6.5 and 6.8. A pseudo-code description of the computation of $\omega_j(\boldsymbol{x})$ with the PH-MNN model is provided in Algorithm 2. A detailed description of the structure of the model is provided in Figure 6.1.

Algorithm 2 Computation of the hazard ratio (ω_j) in the nested PH-MNN model for joint replacement outcome estimation.

Require: $\boldsymbol{\psi}(\boldsymbol{x})$ (output of the neural network block of the model)
Require: $\nu_k^{[age]}(age)$ (natural cubic splines basis functions for age with knots 30,54,68,80,100)
Require: $\nu_k^{[BMI]}(BMI)$ (natural cubic splines basis functions for BMI with knots 15,23,28,36,55)
Require: $\nu_k^{[t]}(t)$ (piecewise linear basis functions for follow-up time with knots 0.2,0.3,1.6,4.3)
Require: $\Theta_{1,j}^{4\times5\times5}$ (parameters)
Require: $\Theta_{2,j}^{3\times4\times4}$ (parameters)
  for j in event types **do**
    $e_{u,j}(age, BMI) \leftarrow \sum_{k,l} \Theta_{1,j}^{u,l,k} \nu_l^{[a]}(age) \nu_k^{[b]}(BMI)$
    $w_{i,n,j}(age, BMI) \leftarrow \text{softmax}(\sum_i \Theta_{2,j}^{m,n,i} e_{i,j}(age, BMI))$
    $\omega_j(t, age, BMI, \boldsymbol{x}) \leftarrow \sum_{i,n} w_{i,n,j}(age, BMI) \nu_n^{[t]}(t) \boldsymbol{\psi}_m(\boldsymbol{x})$
  end for

6.2 Transfer learning for overfit reduction

In order to avoid overfit to be caused by the smaller amount of data available when using preoperative PROMs data as input, we now propose a transfer learning strategy that was made possible by the nested structure described in Figure 6.1. The strategy starts by training the model in the unlinked dataset, which has data from more joint replacements but does not contain PROMs scores. Then the structure of the neural network $\psi_{ij}(\mathbf{x})$ is modified to also include PROMs scores as inputs. This modified model can be split into two components: $\lambda_{i,j}(BMI, age, t)$ and $\psi_{ij}(\mathbf{x})$. There, $\lambda_{ij}(BMI, age, t)$ are the different modes (i) in which the hazard ratio can depend on BMI, age and t for each competing risk j. $\psi_{ij}(\mathbf{x})$ are the set of weights for each mode and each competing risk as a function of the remaining input variables. Although the set of input variables is different in the linked and unlinked datasets, this difference only affects the neural network component of the model: $\psi(\mathbf{x})$. The structure basis function component of the model, $\lambda_{ij}(BMI, age, t)$ is not affected by the increased number of inputs and can be transferred from one model to the other. The parameters of $\lambda_{ij}(BMI, age, t)$ are then fixed to be the same as estimated in the unlinked dataset and the linked dataset is used only to train the neural network component of the model. Moreover, the existence of different competing risks with heterogeneous underlying patterns makes the training speed different for each competing risks so that during the training process it is possible that the model has already fit one par-

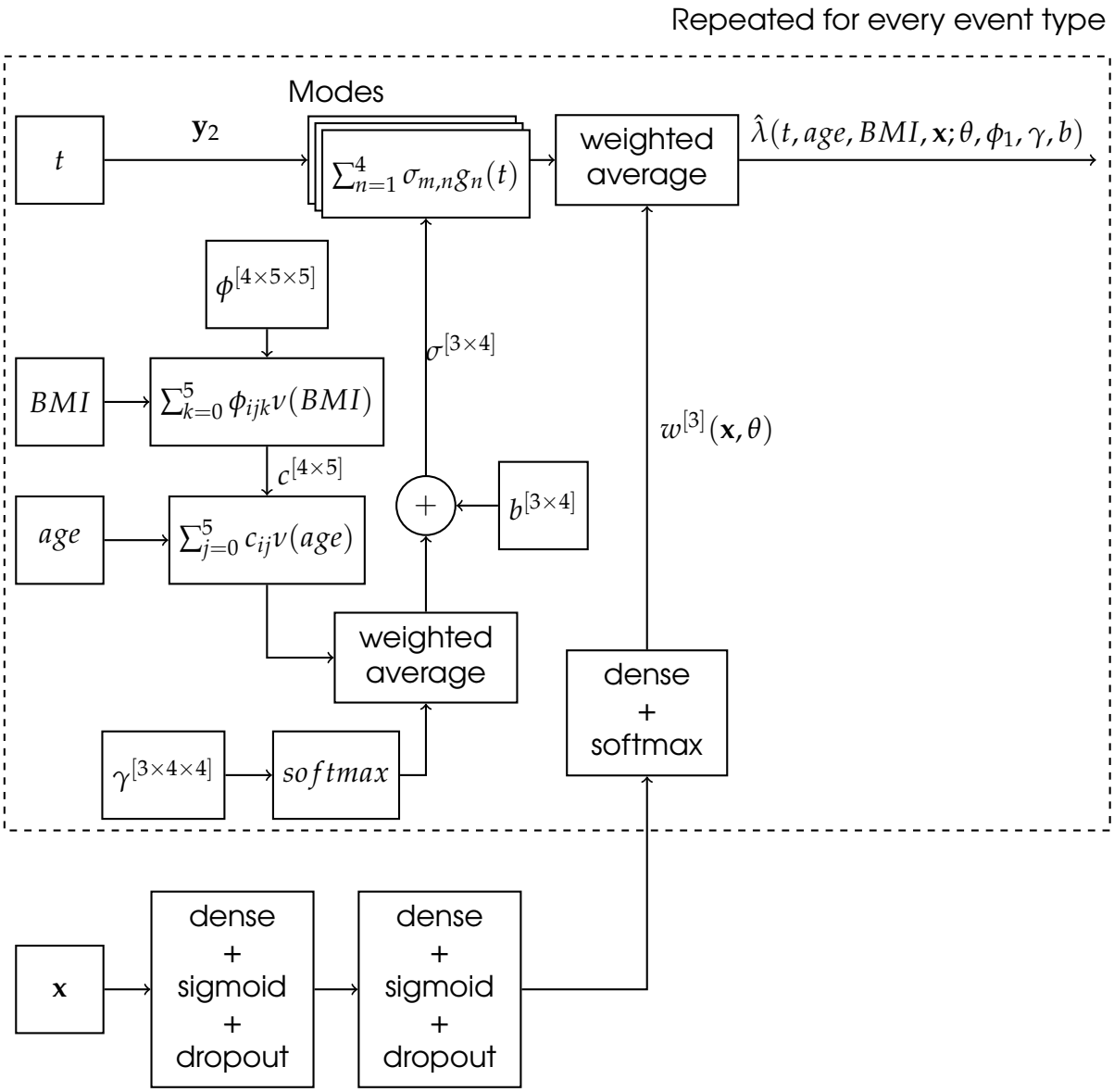


Figure 6.1.: Graphical description of the nested PH-MNN model used in the present chapter.

ticular risk but not the other. To overcome this asymmetry and allow optimum training points for all risks, we increment the transfer learning strategy by stopping the training of the modes $\lambda_{ij}(BMI, age, t)$ at different points depending on the risk j.

With the description of the proposed strategy, it is clear that the scenario where we propose it differs from the standard transfer learning scenario where features extracted with one particular dataset are used as initial parameters to train a model in another dataset with possibly different outcome but with the same structure for the input data. In our case, despite the target outcome being unchanged, the number of inputs changes as a consequence of limited

data availability. This means that the structure of the model needs to change to account for the additional inputs. Nonetheless, we decided to use the same term to designate the strategy because it keeps the core component of the strategy, which is the use of learned parameters from one model in the training of another model. Indeed, we believe that the idea of transfer learning has not been previously used to tackle the problem of learning with missing data precisely because it is challenging to reuse neural network parameters in models with different sets o inputs, and this was made possible with the MNN framework proposed in this work.

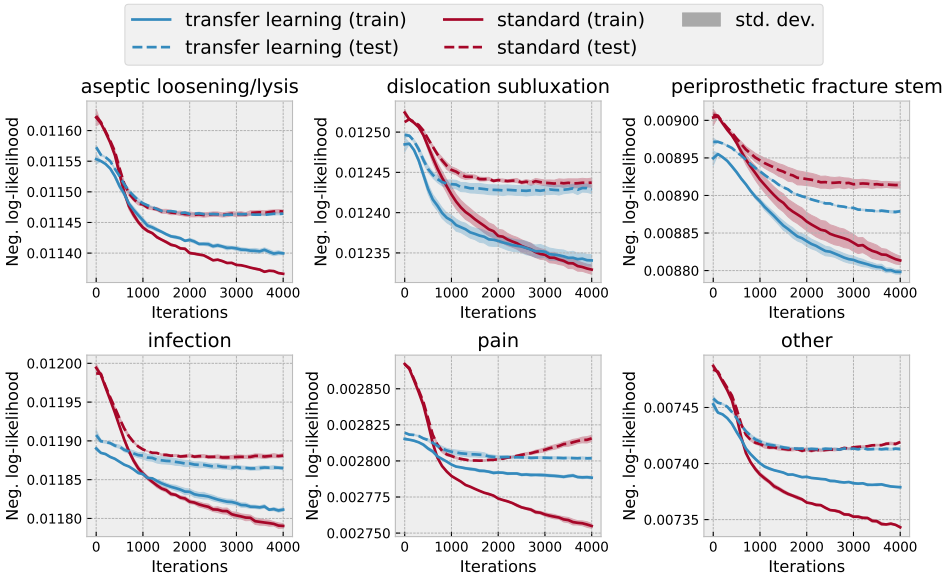


Figure 6.2.: Learning curves comparison of standard and transfer learning approaches in hip replacement revision model.

Figures 6.2, 6.3 and 6.4 show the results of transfer learning procedure in the estimation of revision or mortality after hip or knee replacements. This is done by displaying the training curves with both train and test datasets in two different scenarios: one where transfer learning strategy is used and other where training is performed directly in the linked dataset. The results represent the mean and standard deviation after 10 repetitions of 5-fold cross validation, where the dataset has been randomly permuted before each repetition. In the case of the transfer learning results, the results represent only the training in the linked dataset, after the modes have been fixed. The points where the training of the modes for each competing risk stopped were determined by inspecting the training curves in the unlinked data. This was done using the results of only one split in the first 5-fold cross-validation repetition and the points were kept constant in all instances. In the case of the revision model for knee replacements, it

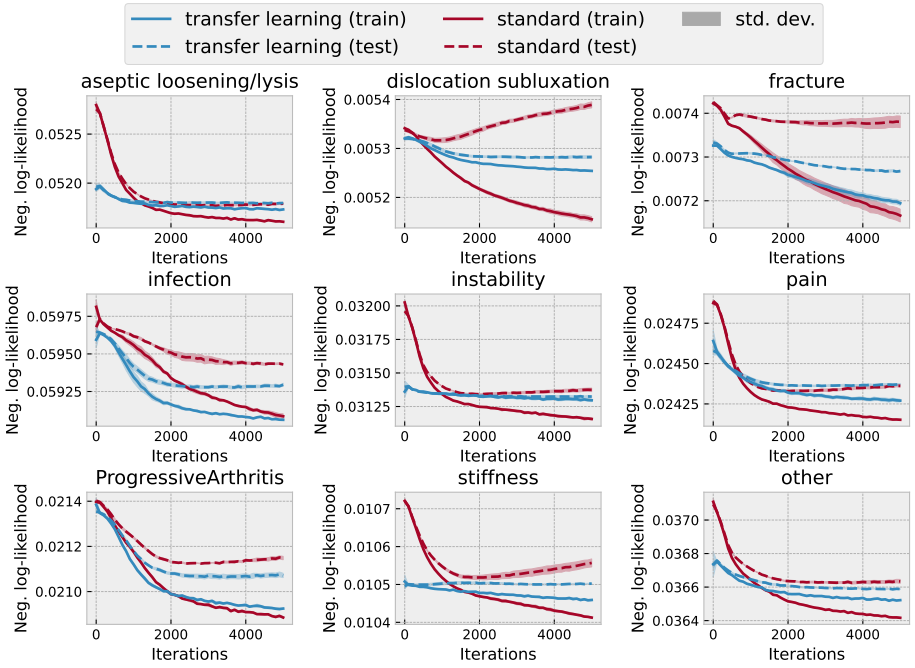
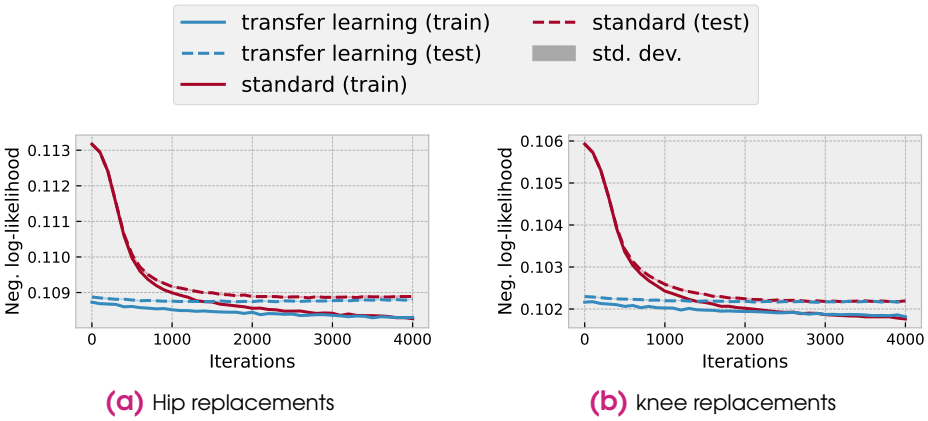


Figure 6.3.: Learning curves comparison of standard and transfer learning approaches in knee replacement revision model.



(a) Hip replacements **(b)** knee replacements

Figure 6.4.: Learning curves comparison of standard and transfer learning approaches in hip and knee replacement mortality model.

was initially observed that fixing the modes to the shapes obtained without the preoperative PROMs inputs resulted in a slight decreased performance for the estimation of instability, pain, progressive arthritis and other. This indicates that introduction of PROMs inputs allowed the distinction of patients within subgroups with that affected the age and BMI dependency of the hazard ratio. Additionally, these risks had a high number of observed events in the linked dataset when compared to other risks in the same dataset, allowing for the modes to be trained without overfit for those particular event types. Our transfer learning strategy allowed a simple solution to this problem which consisted in not

fixing the modes for those event types, using the modes learned in the unlinked dataset only as initial values. Indeed, the hierarchical structure of the nested PH-MNN model has independent heads for each event type, thus allowing the transfer learning strategy to be applied in a modular way only for event types where overfit could be observed. The results displayed in Figure 6.3 reflect the results obtained when not fixing the modes for instability, pain, progressive arthritis and other. The results show that the transfer learning strategy improved the results with a better overall cost and smaller overfitting. The final transfer learning strategy used is summarized in Algorithm 3.

Algorithm 3 Pseudo-code description of transfer learning strategy.

Require: Dataset $\mathcal{D}_{\text{PROMs,train}}$ with PROMs data and $N_{\text{PROMs,train}}$ subjects
Require: Dataset $\mathcal{D}_{\text{PROMs,test}}$ with PROMs data and $N_{\text{PROMs,test}}$ subjects
Require: $\boldsymbol{\psi}_{\text{PROMS}}(\boldsymbol{x}_{\text{PROMs}})$ (neural network block of the model with PROMs input)
Require: Dataset $\mathcal{D}_{\text{train}}$ without PROMs data and N_{train} subjects (containing all subjects in $\mathcal{D}_{\text{PROMs,train}}$)
Require: Dataset $\mathcal{D}_{\text{test}}$ without PROMs data and N_{test} subjects (containing all subjects in $\mathcal{D}_{\text{PROMs,test}}$)
Require: $\boldsymbol{\psi}(\boldsymbol{x})$ (neural network block of the model without PROMs input)
Require: $\Theta_{1,j}^{4\times5\times5}$, $\Theta_{2,j}^{3\times4\times4}$ (parameters)
Require: n_p (baseline number of training iterations with $\mathcal{D}$)
Require: n_{TL} (baseline number of training iterations with $\mathcal{D}_{\text{TL}}$)
$\Theta_{1,j}^{[\text{ref}]}$, $\Theta_{2,j}^{[\text{ref}]}$, $\boldsymbol{\psi}(\boldsymbol{x}) \leftarrow$ MLE using $\mathcal{D}_{\text{PROMs,train}}$ with n_{TL} iterations.
$\Theta_{1,j}$, $\Theta_{2,j}$, $\boldsymbol{\psi}(\boldsymbol{x}) \leftarrow$ MLE using $\mathcal{D}_{\text{train}}$ with n_p iterations
for j in risk types **do**
  if overfit observed in training history for risk j **then**
    $n_{p,j} \leftarrow$ Number of iterations bofore overfit (through visual inspection)
  else
    $n_{p,j} \leftarrow n_p$
  end if
end for
$\Theta_{1,j}^{[\text{TL}]}$, $\Theta_{2,j}^{[\text{TL}]} \leftarrow$ value from iteration n_j of MLE
$\boldsymbol{\psi}^{[\text{TL}]}(\boldsymbol{x}) \leftarrow$ MLE using $\mathcal{D}_{\text{PROMs,train}}$ with $\Theta_{1,j}^{[\text{TL}]}$ and $\Theta_{2,j}^{[\text{TL}]}$ constant for every j.
Repeat MLE with $\mathcal{D}_{\text{PROMs,train}}$, but also optimizing $\Theta_{1,j}^{[\text{TL}]}$, $\Theta_{2,j}^{[\text{TL}]}$ for every j with which the test likelihood with $\Theta_{1,j}^{[\text{ref}]}$, $\Theta_{2,j}^{[\text{ref}]}$ and $\boldsymbol{\psi}(\boldsymbol{x})$ is better than with $\Theta_{1,j}^{[\text{TL}]}$, $\Theta_{2,j}^{[\text{TL}]}$ and $\boldsymbol{\psi}^{[\text{TL}]}(\boldsymbol{x})$.

6.3 Analysis of the resultant models

6.3.1 Hip revision model

The concordance index results for the resulting model are displayed in Table 6.3. Despite the high concordance index for independent event types, when the estimations are aggregated into a single risk estimation, the concordance index becomes considerably lower. This can be explained by the opposing trends observed for different event types. It is noted that the concordance index results do not reflect the overfit observed in the training curves in Figure 6.2. This is consistent with the fact that the concordance index only measure the correct ordering among different estimations but not the calibration of the estimation, so it is possible for a model to achieve perfect concordance index and at the same time estimate survival functions that are completely different from what is observed in the data. Nonetheless, the reduced concordance index for event type "other" with the transfer learning strategy is an indicator that despite the reduction in overfit, there might be aspects of the relationship between the preoperative PROMs and the revision risk that could be not be captured by the model because of the fixation of the age and BMI modes.

Although it is not possible to visualize all variable relationships captured by the model, we provide an illustration of them in Figure 6.5 by showing how each input variable affects the estimation when all other variables are fixed to reference values. These reference values were chosen to be the median within the dataset for numerical variables and the mode for categorical variables. The results show that the same input had sometimes opposing effects to different types of risk. For example, fracture and dislocation/subluxation were more common

Table 6.3.: Concordance index at 8 years follow up time for hip revision estimation with the nested PH-MNN model for individual risks and aggregated single risk.

	nested PH-MNN (standard)	**nested PH-MNN (transfer learning)**
aseptic loosening/lysis	$\mathbf{0.664 \pm 0.003}$	0.662 ± 0.004
dislocation/subluxation	0.612 ± 0.011	$\mathbf{0.627 \pm 0.006}$
fracture	0.635 ± 0.010	$\mathbf{0.649 \pm 0.006}$
infection	0.633 ± 0.005	$\mathbf{0.637 \pm 0.003}$
pain	$\mathbf{0.740 \pm 0.006}$	0.738 ± 0.004
other	$\mathbf{0.660 \pm 0.002}$	0.651 ± 0.002
single risk	$\mathbf{0.577 \pm 0.002}$	0.575 ± 0.002

for elderly patients, with slight increase also for younger patients. Infection risks were the lowest between 60 and 70 years, having an slight increase for both younger and older patients. Aseptic loosening/lysis, pain and other risks were higher for younger patients. This difference in behavior for different event types was also observed for other input variables and explains why the concordance index was lower for single risk than for competing risks.

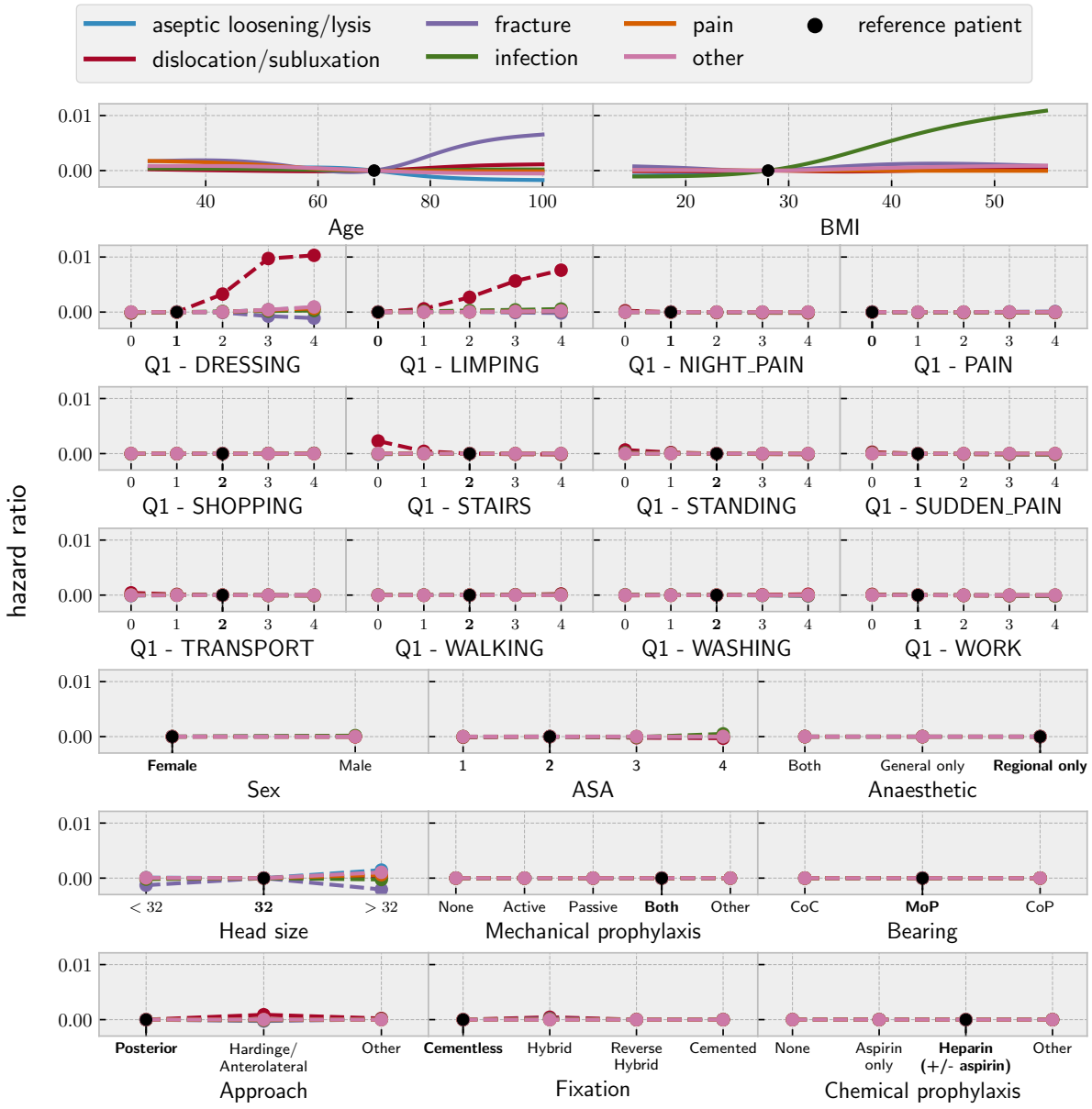


Figure 6.5.: Sensitivity to each input variable for a reference patient int the hip revision model when estimating the hazard ratio with a follow up time of 8 years.

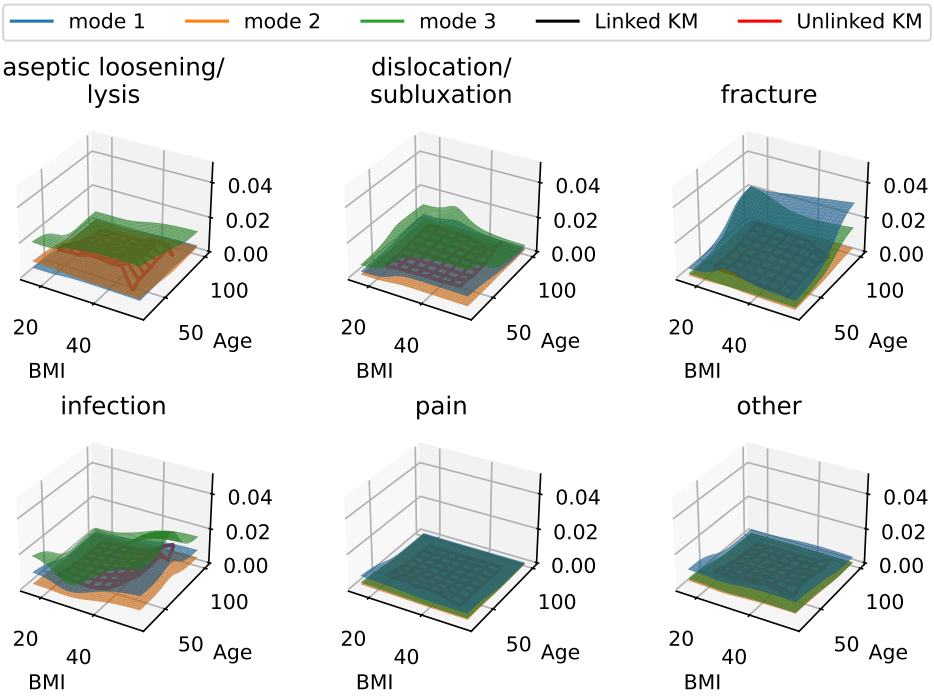


Figure 6.6.: Comparison between age and BMI modes captured by the nested PH-MNN model with the transfer learning strategy and stratified Kaplan-Meier for the hip revision data.

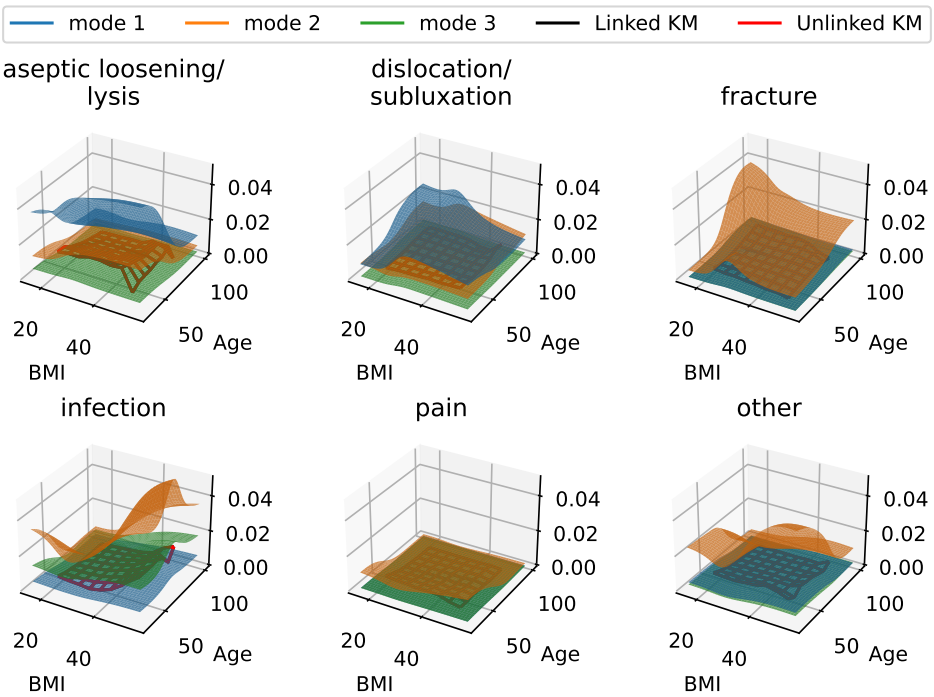


Figure 6.7.: Comparison between age and BMI modes captured by the nested PH-MNN model without the transfer learning strategy and stratified Kaplan-Meier for the hip revision data.

The nested structure in the PH-MNN model also allows a complete visualization of the modes according to which the hazard ratio can vary with age and BMI. Figure 6.7 compares the modes with the Kaplan-Meier estimation stratified by age and BMI when using the transfer learning strategy. Figure 6.6 shows the same comparison when the transfer learning strategy is not used. It is possible to see that Figure 6.7 includes patterns that are not observed in the Kaplan-Meier estimation, which are likely a result of overfit.

6.3.2 Knee revision model

Table 6.4.: Concordance index at 8 years follow up time for knee revision estimation with the nested PH-MNN model for individual risks and aggregated single risk.

	nested PH-MNN (standard)	**nested PH-MNN (transfer learning)**
aseptic loosening/lysis	$\mathbf{0.695 \pm 0.002}$	0.694 ± 0.002
dislocation/subluxation	0.633 ± 0.007	$\mathbf{0.645 \pm 0.005}$
fracture	0.645 ± 0.019	$\mathbf{0.653 \pm 0.011}$
infection	0.598 ± 0.007	$\mathbf{0.612 \pm 0.002}$
instability	$\mathbf{0.698 \pm 0.003}$	$\mathbf{0.698 \pm 0.004}$
pain	$\mathbf{0.705 \pm 0.001}$	0.704 ± 0.001
progressive arthritis	0.623 ± 0.007	$\mathbf{0.634 \pm 0.008}$
stiffness	0.687 ± 0.003	$\mathbf{0.690 \pm 0.003}$
other	0.647 ± 0.002	$\mathbf{0.648 \pm 0.002}$
single risk	$\mathbf{0.638 \pm 0.003}$	0.637 ± 0.002

The concordance index results for the knee revision model are displayed in Table 6.4. Here the single risk concordance index is higher than in the hip revision model, despite being still lower than the concordance index for most individual competing risks. This indicates a higher concordance between risk factors for different event types than in the hip revision model. Here, the concordance index was either equivalent or better for the transfer learning model. As highlighted in the case of the hip revision model, this is not enough to guarantee a better performance for the transfer learning model, but it is an indication that the overfit without the transfer learning strategy prevented the correct ordering of estimations for some of the event types.

Figure 6.8 shows the effect of each input variable to the hazard ratio when all other variables are fixed to reference values. Similarly to the hip revision analysis, the reference values were chosen to be the median within the dataset for

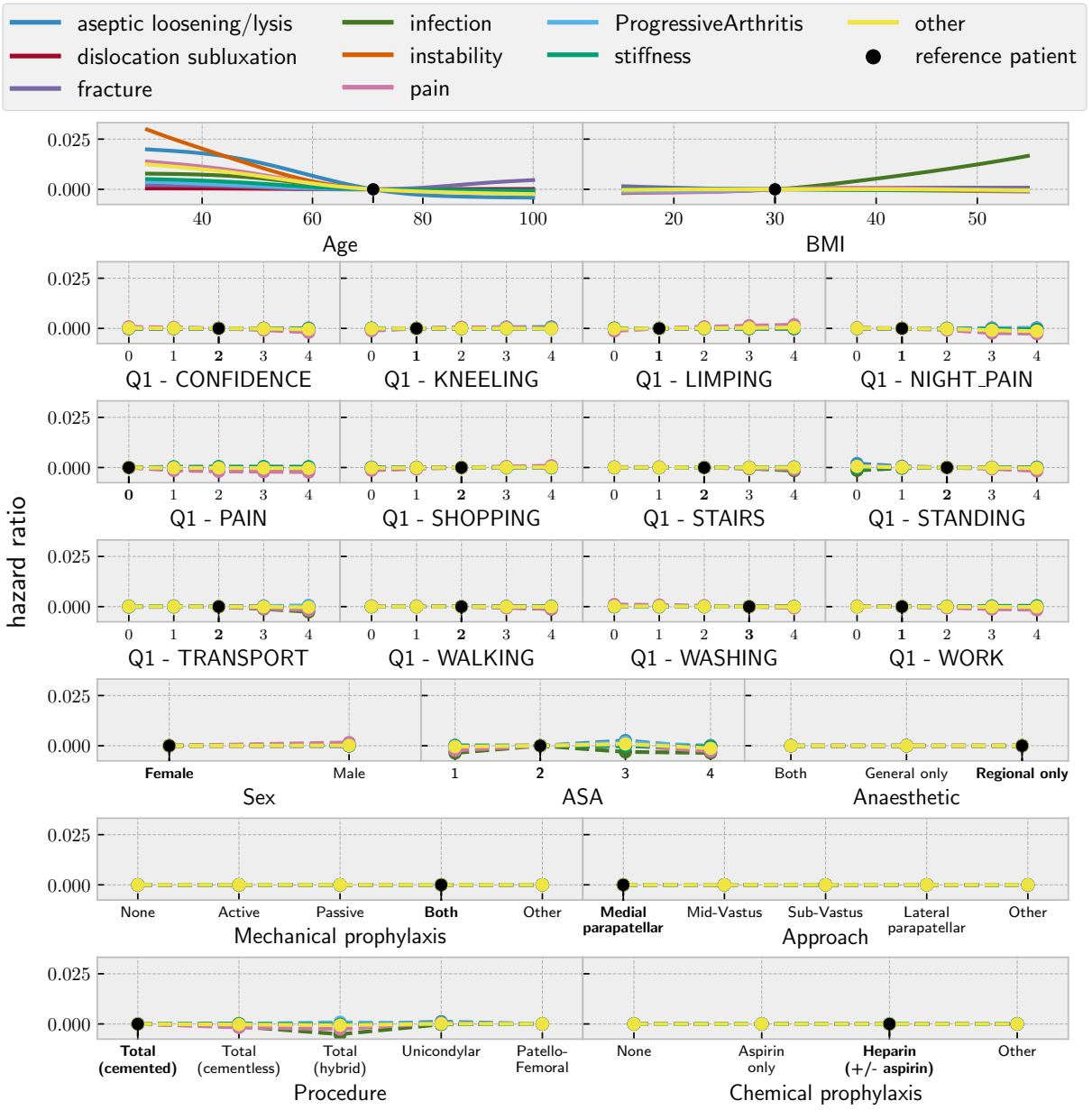


Figure 6.8.: Sensitivity to each input variable for a reference patient int the knee revision model when estimating the hazard ratio with a follow up time of 8 years.

numerical variables and the mode for categorical variables. Despite the single risk concordance index being higher for the knee replacement model than for the hip replacement model, there were also opposing trends observed in the effect of the input variables to the hazard ratio.

Figure 6.9 compares the modes with the Kaplan-Meier estimation stratified by age and BMI when using the transfer learning strategy. Figure 6.10 shows the same comparison when the transfer learning strategy is not used. Both the standard and transfer learning versions of the model presented modes that dif-

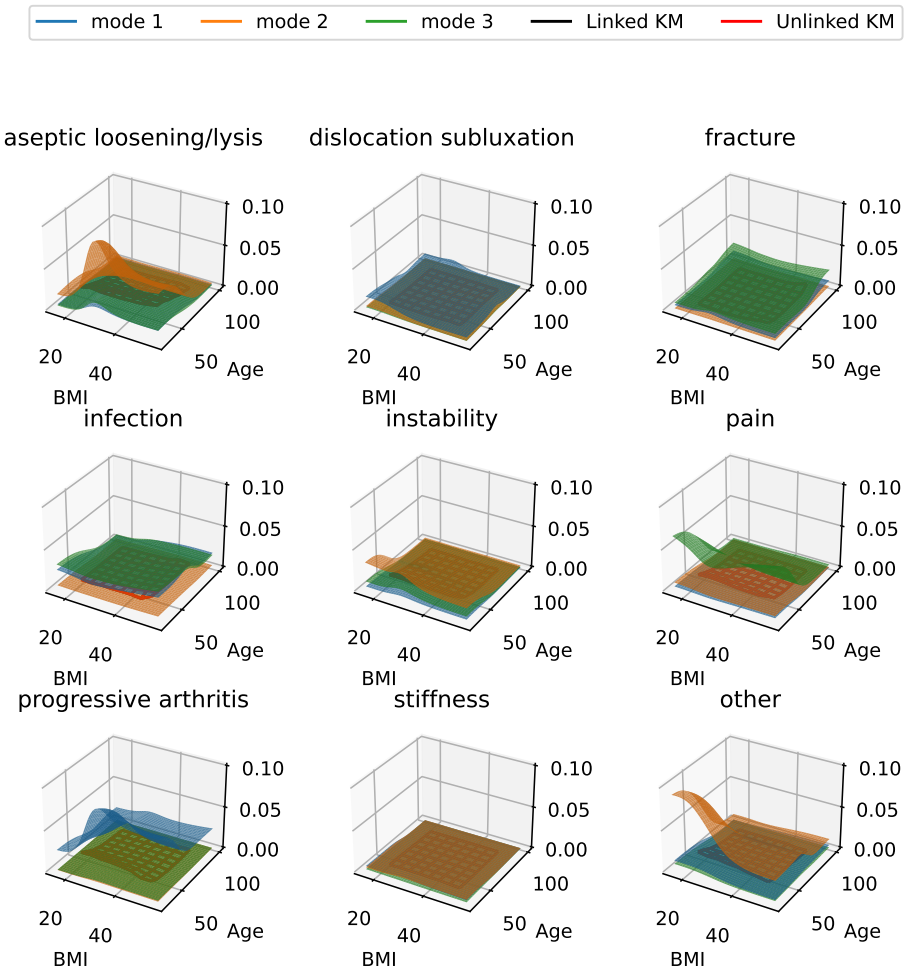


Figure 6.9.: Comparison between age and BMI modes captured by the nested PH-MNN model with the transfer learning strategy and stratified Kaplan-Meier for the knee revision data.

fer substantially from the Kaplan-Meier estimates for the average population, despite the difference being slightly smaller in the case of the transfer learning approach. Nonetheless, the learning curves in 6.3 indicates that this is likely not caused by overfit. One possible explanation for the observed difference is that the knee replacement dataset includes three types of surgery: Total, Unicondylar and Patello-Femoral. Therefore, there is likely a greater variety of response patterns. Indeed, the discordance between the learned modes and the Kaplan-Meier does not necessarily mean that the modes are wrong, since the Kaplan-Meier plot only shows the overall trends among the entire population and the trends can differ among subgroups.

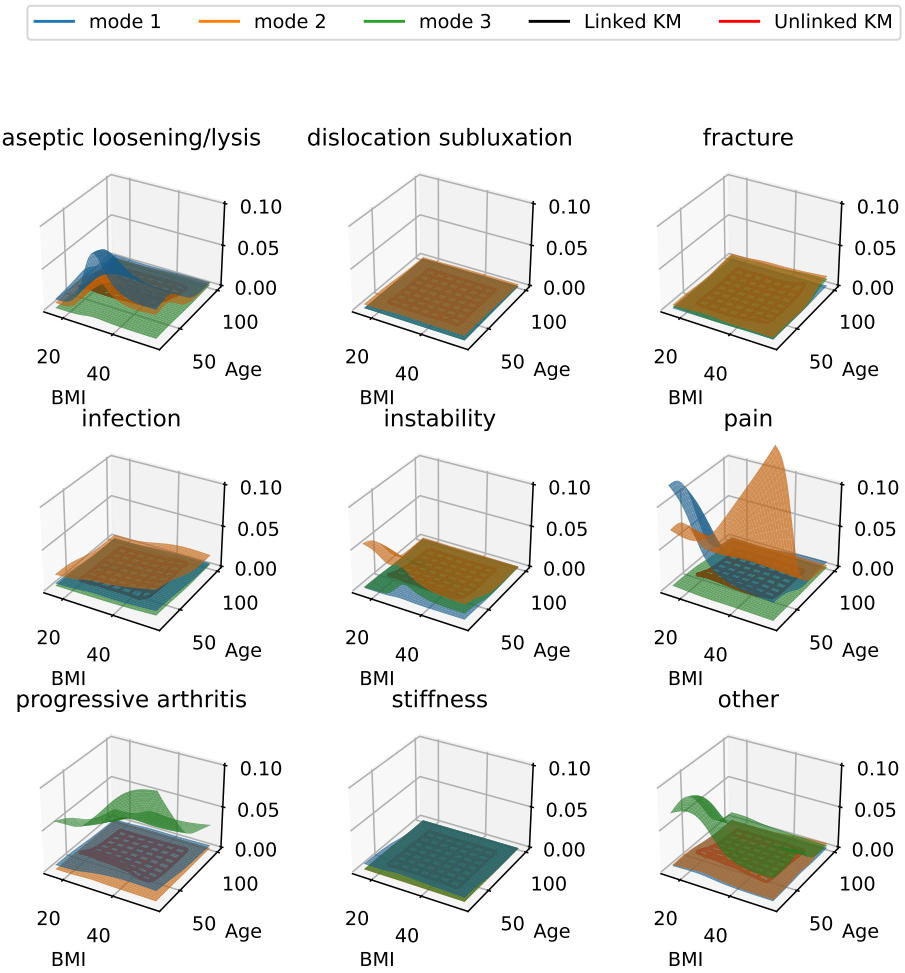


Figure 6.10.: Comparison between age and BMI modes captured by the nested PH-MNN model without the transfer learning strategy and stratified Kaplan-Meier for the knee revision data.

6.3.3 Mortality models

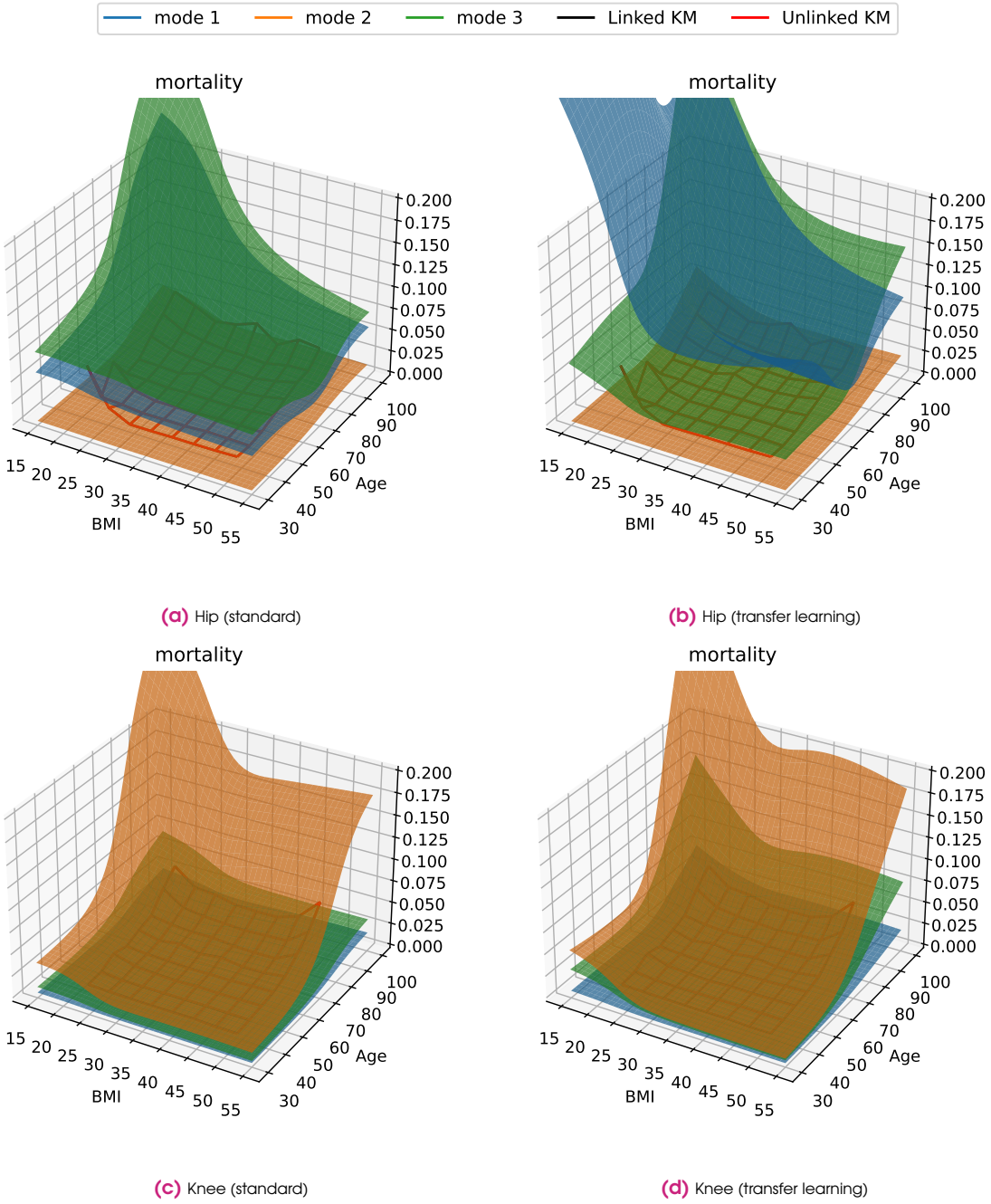
The concordance index results for both hip and knee mortality models are presented in Table 6.5. In both cases the single risk model obtained a high concordance index, which is likely caused by the strong correlation between age and mortality. In both hip and knee models, the concordance index for both

Table 6.5.: Concordance index at 1 year follow up time after hip or knee mortality estimation.

	nested PH-MNN (standard)	**nested PH-MNN (transfer learning)**
hip	0.7638 ± 0.0005	0.7631 ± 0.0008
knee	0.7500 ± 0.0003	0.7497 ± 0.0003

models was equal for both standard and transfer learning training approaches, which is consistent with the fact there was enough data to train the model without the transfer learning approach and overfit was not observed in Figure 6.4.

The modes for the mortality dependency on age and BMI after hip or knee replacement surgeries are shown in Figure 6.11. Here, the transfer learning model



(a) Hip (standard)

(b) Hip (transfer learning)

(c) Knee (standard)

(d) Knee (transfer learning)

Figure 6.11.: Comparison of standard and transfer learning modes of dependency on age and BMI.

for hip mortality had one of the modes that deviates from the Kaplan-Meier observation. Figure 6.4 indicates that this is not caused by overfit. Possible causes include the existence of a pattern that happens only for a small subset of patients that could only be captured by the model with more data available. Another possible cause is that the mode might only appear with a small weight so that no patient would have a pattern similar to it but it would only case a small alteration in the modes with larger weights.

The sensitivity plots in Figures 6.12 and 6.13 show how each input variable af-

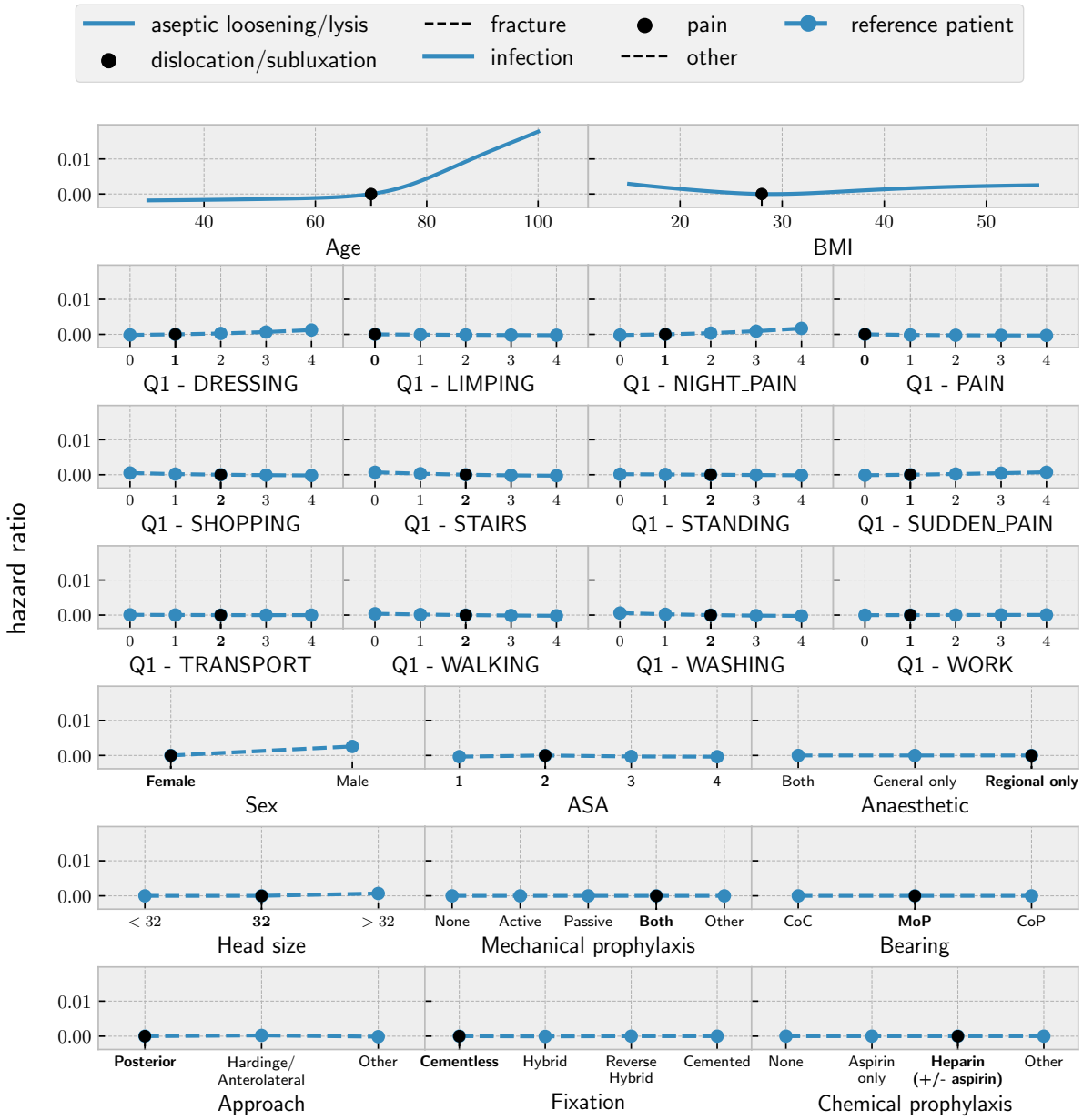


Figure 6.12.: Sensitivity to each input variable for a reference patient int the hip mortality model when estimating the hazard ratio with a follow up time of 1 year.

fects the mortality risk for a reference patient. The figures confirm that the main predictor for mortality is age.

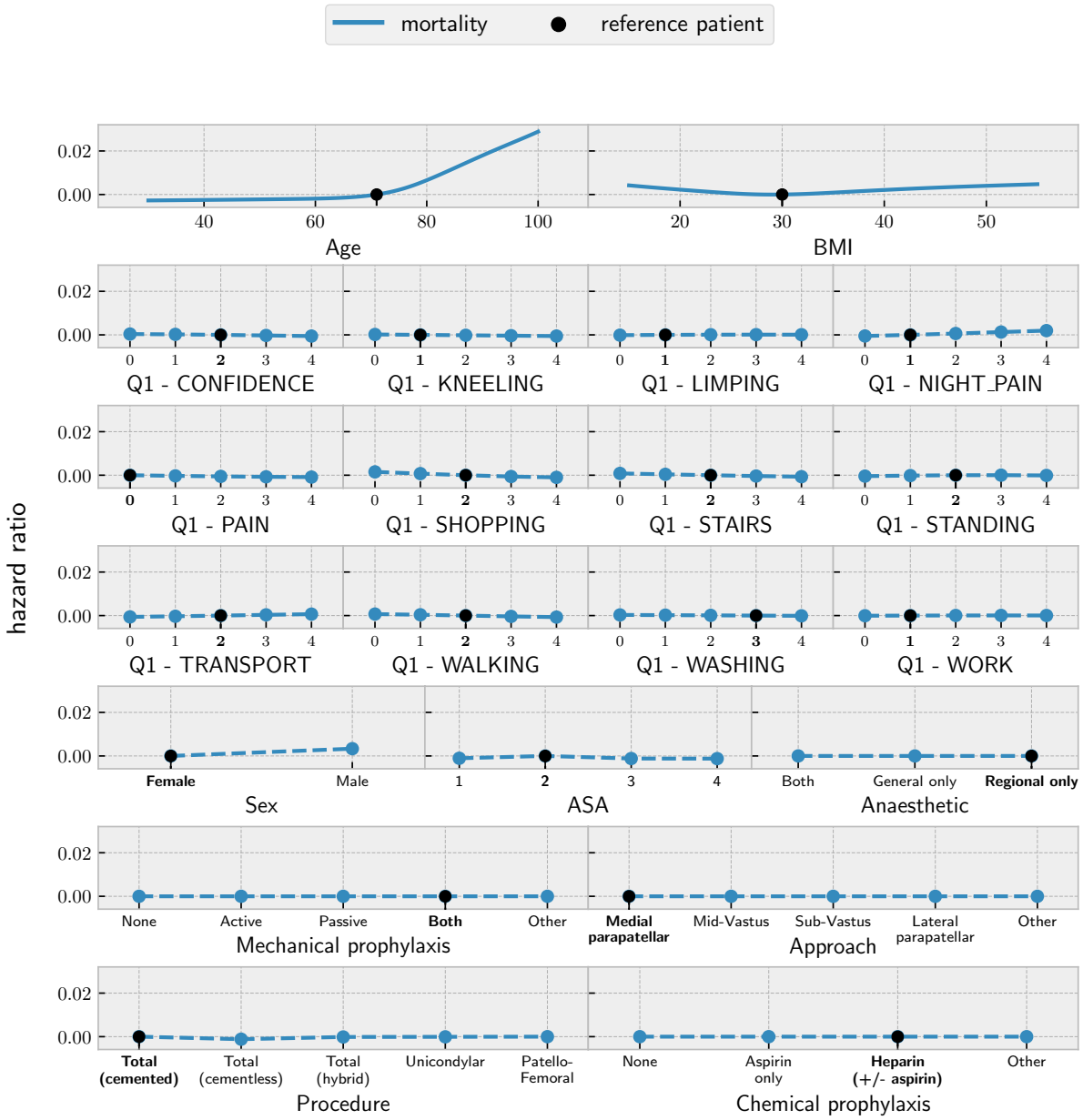


Figure 6.13.: Sensitivity to each input variable for a reference patient int the knee mortality model when estimating the hazard ratio with a follow up time of 1 year.

7

Neural Network Classifier Approach for Postoperative PROMs Prediction

Patient reported outcome measures (PROMs) are numeric evaluations of some aspect of a patient's health. This is obtained through a questionnaire that is answered directly by patients. Examples of PROMs scores that are relevant for hip and knee replacement surgeries include the Oxford hip score (OHS) [89] and Oxford knee score (OKS) [90] that measure pain and functioning of the patient's hip or knee respectively. Also, the EQ5D index and EQ5D VAS [91] provide measurements of the overall health of a patient, which might also be of interest during joint replacement surgeries. The PROMs score change after a surgery is a commonly used metric when evaluating the degree of success of a surgery. The individualized estimation of this change allows the knowledge of how the expected outcome is influenced by the patient's features and other factors that vary from one surgery to another, like prosthesis type and surgical approach. With this information, it is possible to forecast the effect of a surgery prior to it, helping patients to better understand the consequences of a surgery. This facilitates shared decision making for deciding whether or not the patient should undergo a joint replacement surgery, allowing that decision to reflect more the patient's priorities and avoid frustration with the surgery outcome.

Current works on postoperative PROMs estimation make use of standard re-

gression models ranging from linear and Tobit regression [92] to machine learning models [93]. Despite the broad range of regression models that have been applied to this estimation problem, the RMSE obtained by all of them has a similar order of magnitude to the standard deviation of PROMs change in the entire population. This shows that a significant part of the factors that may contribute to the improvement or worsening of a PROMs score after a surgery are not taken into account by these models. Indeed, joint replacement surgeries are highly complex procedures and it would be unreasonable to presume that the demographic information available to the model can fully account for all sources of uncertainty in the surgery outcome. Therefore, without further evidence, it must be presumed that the estimation is inherently uncertain and it is virtually possible that, for any given combination of input attributes, the postoperative score may have any outcome. Therefore, the estimations provided by the models reflect only central tendencies of the score but with no guarantee that the actual outcome will be qualitatively equivalent to estimation. This does not mean that the information provided by the models is useless. Indeed, these models show that there are factors that significantly influence the postoperative PROMs score [92], so the question is not if this information is useful to patients but how to best convey it to them so that it will not lead to unrealistic expectations.

Standard regression models that have been applied to PROMs estimation do have a probabilistic formulation. Indeed, these models use least squared error estimation, which is based on the assumption that the outcome is given by a function of the input plus a Gaussian noise that is independent from the inputs. However, current works do not put noise in evidence and do not test the hypothesis of input independent Gaussian noise. The only model with non-Gaussian noise that has been applied for PROMs estimation was naïve Bayes [93]. However, it relies on strong variable independence assumptions, being not as flexible as neural networks or gradient boosting models. Indeed, naïve Bayes was not the best performing model in the study where it was applied. One difference approach for dealing with uncertainty in PROMs estimation is given by [94]. Instead of modeling the expected value of the outcome, the target is the probability of the PROMs change exceeding a certain threshold. There, no restrictive hypothesis is made to the data being modeled. This is a good alternative for giving patients some notion of the outcome uncertainty, but this probability alone does not provide a full picture of the estimated surgery outcome and it would be better to have a model that combines the estimation of both the expected

postoperative PROMs and the probability of meeting any desired success criterium, like not having a perceivable worsening or achieving an improvement greater or equal to the patient's goal.

A further difference between the present problem and standard regression is that the outcome is restricted not only to a specific range of values. The way in which PROMs questionnaires are built impose that there is a limited number of possible values for its outcome, as will be further detailed in Section 7.1.1. This makes the problem structurally closer to a classification than to a regression problem. We propose the application of a classifier neural network to estimate postoperative PROMs score and show how it can be used to achieve intuitive interpretations to the expected outcome of a joint replacement surgery. Specifically, we show that this model can not only estimate the expected value of postoperative PROMs score with better performance than current state of the art, but also provide the probability of the result being within some interval of interest. This information is particularly useful for intuitively conveying uncertainty information to patients. For example, it is possible to estimate the probabilities of the postoperative score being higher, lower or equivalent to the preoperative score. Also, if the patient or surgeon has a specific goal for the postoperative score, it is possible to accurately estimate the personalized probability of the score being higher or equal to this goal.

7.1 Model Formulation

7.1.1 Data properties and estimation uncertainty

The Oxford Hip Score (OHS) and the Oxford Knee Score (OKS) are both obtained from the answers to 12 independent questions made to patients about their joint health. Although the questions are different for the OHS and the OKS and their numerical values can have different interpretations, both restrict the answer to each question as an integer value from 0 to 4, so the total score must be an integer from 0 to 48.

The EQ5D questionnaire is composed of 5 question about different aspects of a patient's health with answers being an integer number from 1 to 3 and the VAS, which is an integer from 0 to 100 that is directly reported by the patient as their perception of their overall health. The first group of questions is used either

as a list of all answers or as an index, which maps each possible combination to a real number. The question can have in total $3^5 = 273$ possible answers and consequently the possible values for the index despite being represented by real numbers is finite. The coefficients used in the index computation vary from country to country depending on various factors.

Why are current models poorly suited to outcome estimation?

The estimation of these scores is traditionally performed in the domain of regression models, but standard regression models are not perfectly applicable since they assume a scenario where the outcome has an infinite range. In principle, the pain and functioning of some joint and the broader patient health as measured by the EQ5D could all be viewed as having an infinite range, but the PROMs questionnaires can only measure it within a limited range of the possible outcomes because of the "floor" and "ceiling" effects.

Current solutions to this problem include the use of Tobit [95] and tree type [96] models, that restrict the estimated outcome to the range that is actually possible for the metric of interest. In the Tobit model, this is achieved by clipping the output of a standard regression model and adapting the likelihood function to account for that change. In tree type models, the estimated outcome is the weighted average of real output data from patients with similar characteristics to the target patient. Since the estimated outcome will always be some type of average of the outcomes of real patients, it will always be in the allowed range. Both these methods successfully deal with the restrictions in the output data range, but they allow the estimation to assume arbitrary non-integer values. This is coherent with the definitions of the PROMs scores because even with the measurements restricted to a finite set of values, their expected value can still assume intermediary values.

As detailed in the beginning of the present chapter, results from current state of the art regression models are not enough to guarantee that the estimated outcome will be qualitatively equivalent to the actual observed outcome. This is perfectly coherent with the phenomenon being modeled since joint replacement surgeries are complex procedures whose outcome cannot be completely determined beforehand. Nonetheless, this can easily lead to false expectations when the estimation is provided to patients as a single point value. Indeed,

the RMSE observed in existing PROMs models has the same order of magnitude as the standard deviation of the change score in the entire population, which means that from all factors that influence the final outcome, the ones that are not captured by the model have a similar influence to those captured by the model. In principle, this could be either because the input variables used in estimation are not enough to determine the surgery outcome or because the measurement noise in the PROMs questionnaire makes the RMSE high despite the underlying phenomenon being accurately measured.

This can be better understood if we qualitatively split the PROMs signal according to two different criteria: first, between the component that is caused by the joint pain and functioning and the one that is caused by the patient's perception of it; second, these two groups can be further split into low and high frequency components. The low frequency component of the joint pain and functioning is the actual phenomenon that we aim at studying and the high frequency component of it can be regarded as noise. Regarding the component that is due to the patient's perception of pain and functioning, although perception differences are not directly caused by a problem in the joint, the same amount of pain might be tolerable to some patients and not to others. Given that the joint replacement surgery aims at treating a specific patient, it is reasonable to take the patient's perception into account in evaluating the degree of success of that surgery. The patient's perception can also be split into low and high frequency components and the low frequency component is the one that is caused by the surgery, while the high frequency component can be regarded as noise.

The models themselves are not capable of distinguishing between different components of the signal, even between low and high frequency components, since there is no data available about it. However, other experimental settings have been used to estimate high frequency variation of OHS/OKS [97, 98]. There, multiple questionnaires answered by the same patient are used to show how the answer varies with time despite the patient not receiving any treatment that should alter the joint health. This measurement only takes into account the short term variability of both the joint health and the patient's perception of it. If the uncertainty in the models were caused only by this factor, the RMSE would be considerably lower than what is observed in estimation models. Since the data available does not contain short term repetitions of the same questionnaire, which would be necessary to reduce the high frequency noise, we

	nonlinearity	floor/ceiling effects	uncertainty
Linear regression	✗	✗	✗
Tobit regression	✗	✓	✗
Regression NN	✓	✗	✗
Regression XGB	✓	✓	✗
Classifier NN	✓	✓	✓

Table 7.1.: Qualitative evaluation of postoperative PROMs estimation models according to the capacity of representing three different aspects of the target function: nonlinearity, floor/ceiling effects and uncertainty.

include it as part of the uncertainty of the model, which is adequate given that the low frequency noise is substantially higher than the high frequency noise.

This analysis makes it clear that the postoperative PROMs score would be best estimated as a probability distribution instead of a single point estimate as it is usually done. The major problem with estimating a probability distribution is that the result is neither intuitive nor directly useful to patients, but it can be used to achieve more intuitive measures. For example, the single point estimation can be retrieved from it by computing the expected value or median of the outcome probability distribution. Although modeling the outcome as real valued is coherent for single point estimate models, restricting the outcome to only the finite set of numbers that can actually be obtained in the questionnaires is the best option when estimating the full probability distribution of the postoperative score. Indeed, this restriction allows the probability distribution to be modeled with standard machine learning classification models. There, each possible value for the outcome is considered an entirely different outcome, and the model estimates for each patient the probability of each of these outcomes. This set of probabilities contains the full specification of the outcome probability distribution since all possible outcomes of the postoperative PROMs questionnaire are accounted for.

7.1.2 Model Structure and Estimation

The model used in this work is a classification neural network [99] with sigmoid activation function, dropout and a softmax activation function in the visible layer. The likelihood of this model is given by the categorical cross entropy. Estimation of the parameters is performed through likelihood maximization with the Adam optimizer [100].

7.1.3 Interpretability of the Outcome of the Model

The probabilities of each possible score as estimated by the model might not provide by themselves an intuitive estimation of the surgery outcome. However, this set of probabilities is the full probabilistic description of the target score outcome and if accurately estimated, it will contain all the information that can be obtained about the postoperative outcome given the available input variables. Consequently, all other meaningful information that can be obtained by other models can also be extracted from this estimated probability distribution:

- **Minimum Squared Error Estimation:** The output of most models for individualized postoperative PROMs score estimation corresponds to the minimum squared error estimation. The reason why regression models are trained with this metric is that in the case of Gaussian noise the model will converge to the expected value of the outcome if enough data is provided. Given the probability distribution of the score, the expected value of the outcome can also be obtained. Indeed, it is preferable to estimate the expected value using the outcome probability distribution than to rely on minimum squared error estimation as the later will not be adequate for all types of noise.

- **Minimum Absolute Error Estimation:** Although less common, the minimum absolute error estimate is also a useful outcome for regression models. The reason why regression models are trained with this metric is that in the case of Gaussian noise the model will converge to the median of the outcome probability distribution. Since in our model, the entire probability distribution is estimated, the median of it can also be obtained from it. Once again, estimation directly from the probability distribution is preferable since the minimum absolute error estimate will not converge to the median if the error is unbalanced.

- **Variance**: The variance of the estimated probability distribution is commonly used as an indicator of the accuracy of the estimation. It can be retrieved from the categorical probability distribution as the expected value of the squared outcome minus the square of the expected outcome. Therefore, it can also be extracted from the outcome probability distribution.

- **Confidence Intervals:** A confidence interval is also used as an indicator

of uncertainty. It is defined as an interval within which the measurement will be with some desired confidence probability, usually 95%. In the case of non Gaussian distributions, it describes the uncertainty more accurately than the variance, since the distribution might not be symmetric. This value can also be extracted from the outcome probability distribution, which is done by determining the outcome value for which the cumulative distribution will reach the values of 2.5% and 97.5%.

- **Probability of Specific Intervals:** A particularly useful information that can be obtained from a categorical model is the probability of the outcome being within some specified interval of interest. The probability of the outcome assuming exactly some value might not be of a great practical interest. However, if the range of possible outcomes is divided into intervals with practical interpretation, the probability of the outcome being in each of these intervals will be of great interest. As an example, the range of outcomes can be divided into (i) smaller than the preoperative score and (ii) greater or equal to the preoperative score. The probability of the outcome being in each of these ranges can be easily determined by summing the probabilities of all outcomes in the desired interval.

The possibility of making probabilistic estimations is an important addition to the estimation of the expected outcome. Indeed, the uncertainty in the estimation of the expected outcome cannot be neglected in current models, so it is crucial that the estimation is accompanied by some measurement of uncertainty. This is especially true when the uncertainty depends on the patient attributes, so that the estimation will be more precise for some patients than for others. With this information, it is possible to convey the uncertainty to patients in a manner that is both intuitive and precise. This is achieved through the risk of the postoperative score being lower than the preoperative score.

7.2 Missing Data Imputation

The missing data in the National Joint Registry PROMs dataset can be divided into two groups: patients with no information on the postoperative PROMs score; and patients with missing data in some of the input variables. In the first group, it is not possible to compensate for the missing data and it is necessary to restrict the modeling to patients who have completed the PROMs quaestionaire. In the

second group, there are three possible ways in which the data might be missing: missing completely at random (MCAR); missing at random (MAR); or missing not at random (MNAR) [101]. In either case, the safest approach for modeling is to apply a data imputation technique assuming a MAR scenario. If the data turns out to be MCAR, this approach will be redundant but not harmful. On the other hand, if the data turns out to be MNAR, this approach will not be enough for retrieving all the missing information, but it is the most sensible option with the available data. Indeed, an MNAR hypothesis would mean that there is a bias in the unobserved variables so they tend to be either smaller or larger than the observed ones. To verify this hypothesis, it would be necessary to design a particular experimental setting that would allow measuring some of the missing data. Without additional data the only two possibilities are to assume that there is no bias or to assume a particular bias that is not based on data. Since there is no particular reason to believe that there is a bias, the most sensible assumption is that there is no bias, which is equivalent to saying that the data is MAR.

When only patients with complete postoperative PROMs data are taken into account, most patients with missing data had only data missing for BMI. In the dataset used, the BMI is restricted to integer numbers, which allows the use a classifier neural network with integer BMI values as outputs as the imputation model. Restricting the imputation to only BMI allows the multiple imputations to be performed into one single step as follows. Then, the estimated probability distribution for the imputed model will be given by the average of the estimations for the models trained with each different sample.

Algorithm 4 Multiple imputations procedure for training PROMs with incomplete BMI data.

Require: $N_{steps} > 0$
Require: x_c (set of all input variables except BMI)
  Train the classifier NN imputation model: $p_{BMI}(x_c)$
  $n \leftarrow 0$
  while $n < N_{steps}$ **do**
    sample x_{BMI} from $p_{BMI}(x_c)$ for each patient with missing BMI
    train the n^{th} PROMs model with the imputed BMI data
  end while

7.3 Model Implementation and Validation

7.3.1 Data description

In the present section, we use the neural network classification model to estimate the postoperative Oxford and EQ5D scores after hip and knee replacement surgeries with data from the National Joint Registry of hip and knee replacement surgeries performed in England from 2009 to 2018 linked with PROMs data with a total of 278.655 knee replacements and 249.634 hip replacements. Figures 7.1, 7.2, 7.3 and 7.4 show the distribution of preoperative, postoperative and change PROMs after hip/knee replacement surgeries. Age was restricted from 30 to 100 (14 knee replacements ignored and 170 hip replacements ignored). BMI was restricted from 15 to 55 (358 knee replacements ignored and 239 hip replacements ignored). Hip replacements with bearing types MoM, CoM and MoC were not included in the model since these types of implants are no longer used (5402 hip replacements ignored). Head size in hip replacement surgeries were grouped into <32, 32 and >32, since 32 is the standard size. BMI was missing for 70693 knee replacements and 62907 hip replacements. In total, less than 754 patients had variables other than BMI missing in knee replacements and less than 9.127 patients in hip replacements. Since the majority of missing data has only BMI missing, it was decided to only impute BMI data since it will handle most of missing data while avoiding the need of recursion in the multiple imputation steps as discussed in Section 7.2.

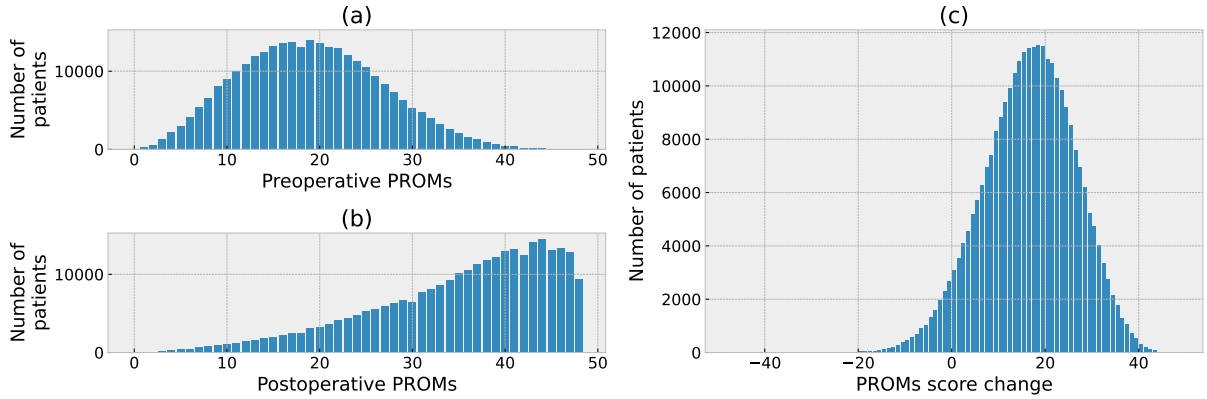


Figure 7.1.: Histogram of the OKS before and after knee replacement surgeries. Panel (a) shows the preoperative OKS; panel (b) shows the postoperative OKS; and panel (c) shows the OKS change score.

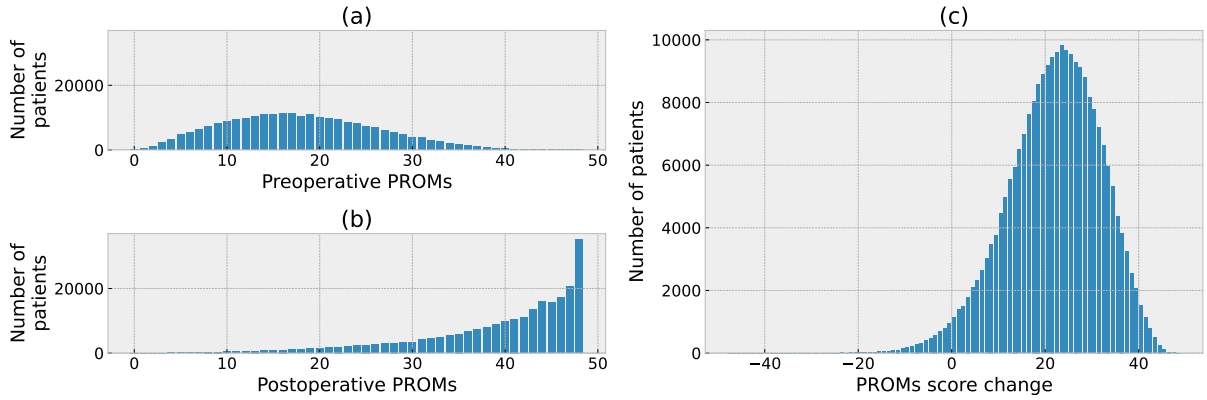


Figure 7.2.: Histogram of the OHS before and after hip replacement surgeries. Panel (a) shows the preoperative OHS; panel (b) shows the postoperative OHS; and panel (c) shows the OHS change score.

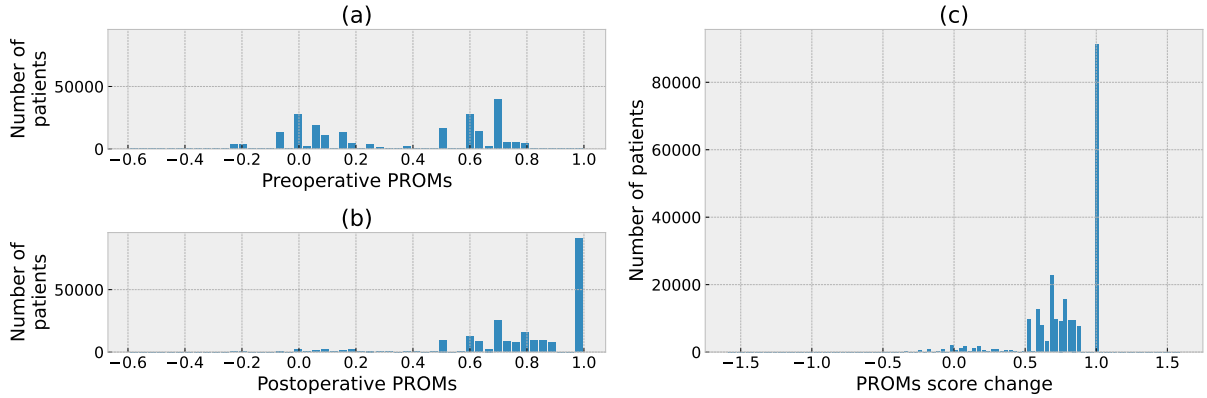


Figure 7.3.: Histogram of the EQ5D index before and after hip replacement surgeries. Panel (a) shows the preoperative Eq5D index; panel (b) shows the postoperative EQ5D index; and panel (c) shows the EQ5D index change score.

7.3.2 Model validation

The estimation was divided into separate models for each combination of joint and target outcome, making a total of 6 models. For knee replacements the targets were OKS, EQ5D index and VAS score. For hip replacements the target were OHS, EQ5D index and VAS score. The input variables that were used were: sex, BMI, age, surgery date, ASA, chemical prophylaxis, mechanical prophylaxis, comorbidities, approach, lead surgeon grade, implant reason, head size (only for hip), procedure type (only for knee), and the individual answers to questions in the preoperative PROMs quaetionnaire. In the case of OHS/OKS target, the input preoperative score was the set of answers (integers from 0 to 4) to each of the question that make the target score. In the case of EQ5D index or VAS score targets, the input preoperative score was composed by both the VAS score (integer from 0 to 100) and the individual answers to the 5 questions on the EQ5D

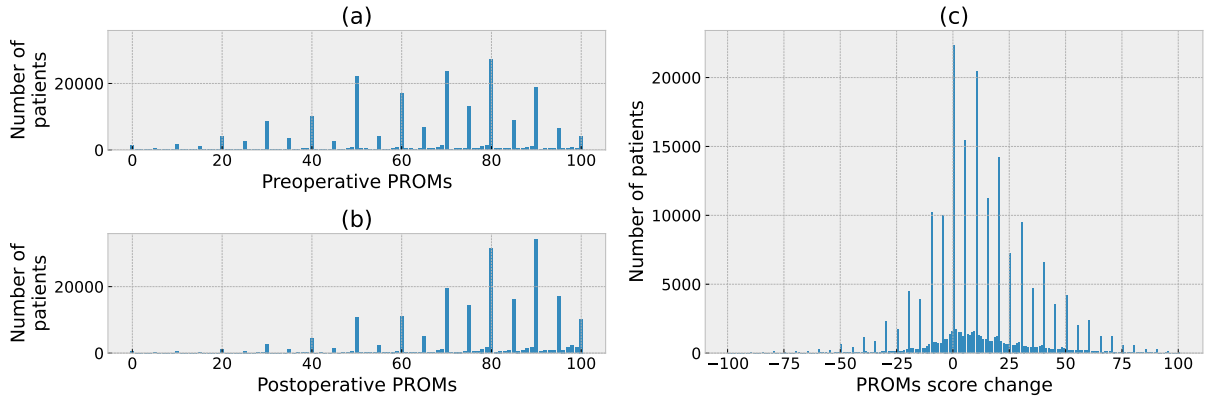


Figure 7.4.: Histogram of the VAS score before and after hip replacement surgeries. Panel (a) shows the preoperative VAS; panel (b) shows the postoperative VAS; and panel (c) shows the VAS change score.

index (integers from 0 to 2). This maximizes the use of the available input information for the model when compared to using only aggregated scores as input. Since each question accounts for a different aspect of either the joint pain and functioning or patient overall health, this allows the evaluation of how each of these aspects influence these postoperative outcomes. Estimations were performed with both the complete data and the imputed BMI versions of the neural network classification model.

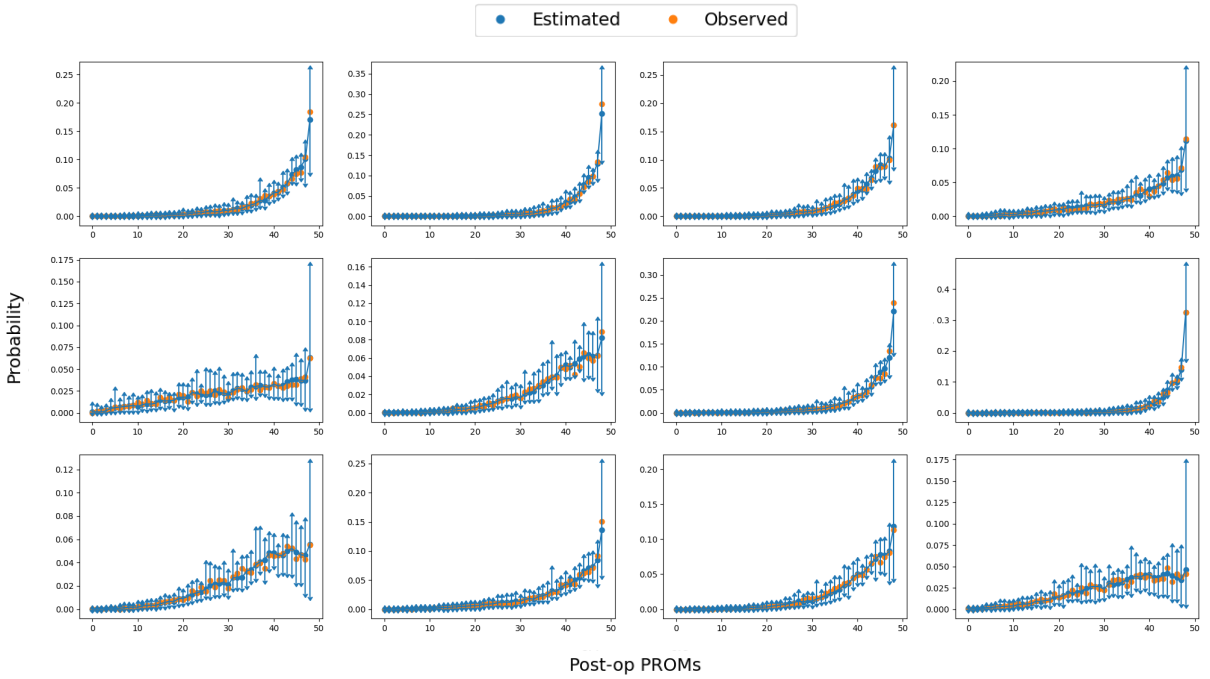


Figure 7.5.: Illustration and measurements of the postoperative OHS after hip replacement divided by clusters in the estimated probability distribution.

The calibration of the model was assessed by grouping the patients into

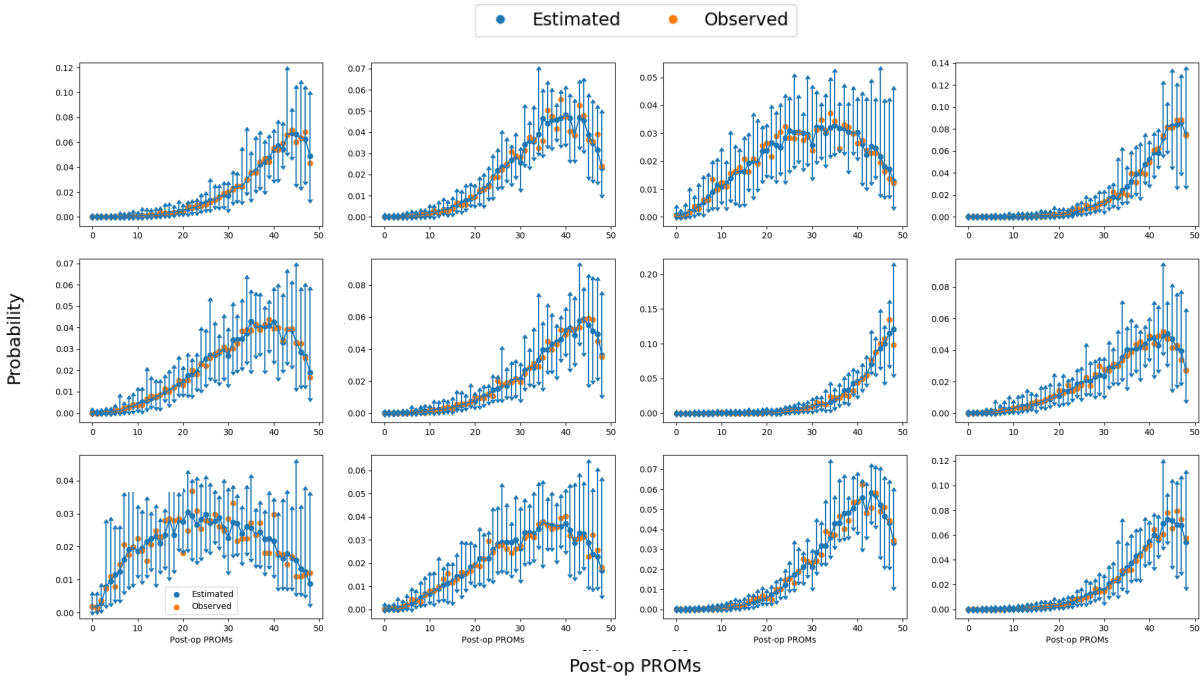


Figure 7.6.: Illustration and measurements of the postoperative OKS after knee replacement divided by clusters in the estimated probability distribution.

clusters using the K-means algorithm with the estimated probability distribution as input. Within each cluster, the expected probability distribution was computed and the results compared to observation. Figures 7.5 and 7.6 show the results from the cluster analysis. The results show that the frequency of each outcome in each cluster was compatible with its estimated probability, which means the probabilities estimated by the model are consistent with the data.

Although the outcome probability distribution refers to the time after surgery, its estimation is performed only with information available previously to surgery. Therefore, the clustering is entirely based on input variables. The results from this calibration analysis shows clearly that the postoperative outcome is always spread throughout a wide range of values. This confirms the argument given in the Introduction for the estimation being highly uncertain and requiring a probabilistic component to provide a realistic estimation of the outcome.

7.4 Model Comparison

The performance of the proposed model was compared with current state of the art models by estimating all of them with the same data cohort and input variables and comparing the resulting evaluation performance. The models

included in the comparison were:

- linear regression [92];
- Tobit regression [92];
- regression neural network [93];
- regression XGB [93];
- classification neural network as proposed in the present work.

Both the regression neural network and the classification neural network has two hidden layers, with 100 units and 0.5 Gaussian dropout in the first hidden layer and with 30 units and 0.2 dropout in the second. The regression XGB model had learning rate 0.3, max depth 4, sub sample 1.0 and 100 estimators. Evaluation was performed both in the data subset with complete data and in the subset where only patients with missing BMI are taken into account. Since imputed data versions of the reference models are not used in the literature, they were trained with only complete data and their estimates for patients with missing BMI are given by the weighted average of their estimation for each possible value of BMI. There, the weights are given by the output of the neural network classification imputation model of the BMI. The evaluation criteria used were the root mean square error (RMSE), the mean absolute error (MAE) and the AUC for the probability of meeting specific score thresholds for both the postoperative PROMs and their change relative to the preoperative PROMs. The RMSE and MAE results are given in Table 7.2 for complete data cohorts and Table 7.3 for imputed BMI cohorts. The AUC for the estimation of PROMs change score is given in Figure 7.7 for hip and in Figure 7.8 for knee replacement surgeries.

Table 7.2.: Complete data results and 95% CI of the root mean square error (RMSE) and mean absolute error (MAE) for the postoperative PROMs score estimation for each model after hip or knee replacement surgeries. Evaluation was performed with the complete data cohort.

	Knee			**Hip**		
	RMSE	MAE	MID AUC	RMSE	MAE	MID AUC
Classifier NN	8.5896 ± 0.0008	6.7356 ± 0.0007	$68.73\% \pm 0.01\%$	7.8594 ± 0.0006	5.7067 ± 0.0008	$73.96\% \pm 0.02\%$
Tobit	8.6497 ± 0.0012	6.8252 ± 0.0011	$66.83\% \pm 0.00\%$	8.1406 ± 0.0028	5.7974 ± 0.0015	$70.98\% \pm 0.00\%$
Regression NN	8.5883 ± 0.0005	6.8688 ± 0.0091	$67.13\% \pm 0.03\%$	7.8583 ± 0.0001	5.9867 ± 0.0080	$70.96\% \pm 0.03\%$
Linear	8.6243 ± 0.0000	6.8860 ± 0.0000	$66.66\% \pm 0.00\%$	7.9015 ± 0.0003	6.0035 ± 0.0005	$70.53\% \pm 0.00\%$
Regression XGB	8.6116 ± 0.0022	6.8652 ± 0.0018	$67.06\% \pm 0.01\%$	7.8880 ± 0.0014	5.9808 ± 0.0015	$70.87\% \pm 0.01\%$

Table 7.3.: Imputed data results and 95% CI of the root mean square error (RMSE) and mean absolute error (MAE) for the postoperative PROMs score estimation for each model after hip or knee replacement surgeries. Evaluation used only data from patients with missing BMI.

	Knee			**Hip**		
	RMSE	MAE	MID AUC	RMSE	MAE	MID AUC
Classifier NN	8.6404 ± 0.0003	6.7872 ± 0.0008	$68.55\% \pm 0.01\%$	7.8896 ± 0.0005	5.7435 ± 0.0004	$73.87\% \pm 0.01\%$
Tobit	8.7009 ± 0.0008	6.8769 ± 0.0006	$66.61\% \pm 0.01\%$	8.1663 ± 0.0004	5.8313 ± 0.0005	$70.69\% \pm 0.00\%$
Regression NN	8.6403 ± 0.0007	6.9102 ± 0.0035	$66.93\% \pm 0.02\%$	7.8875 ± 0.0003	6.0196 ± 0.0021	$70.75\% \pm 0.01\%$
Linear	8.6791 ± 0.0001	6.9378 ± 0.0002	$66.42\% \pm 0.00\%$	7.9347 ± 0.0001	6.0403 ± 0.0003	$70.25\% \pm 0.00\%$
Regression XGB	8.6531 ± 0.0002	6.9081 ± 0.0002	$66.93\% \pm 0.00\%$	7.9043 ± 0.0009	6.0066 ± 0.0012	$70.65\% \pm 0.02\%$

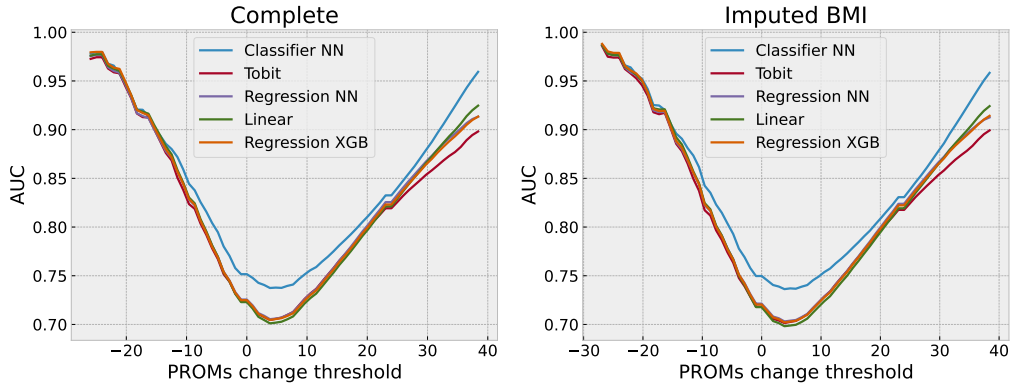


Figure 7.7.: AUC for the capability of the model to measure the probability of the change in PROMs score after a hip replacement surgery to be above a certain value. For each value in the x axis, it is possible to define a threshold and estimate the probability of the OHS variation to exceed this value. The figure gives the resulting AUC (y axis) for the probability of the OHS change to exceed each possible threshold (x axis). Panel (a) shows the results for the complete data cohort and panel (b) shows the results for the missing BMI cohort.

Results in Table 7.2 show that the proposed classifier neural network outperforms all other reference models according to both the RMSE and the MAE. Conversely, the imputed version of the model performs worse and its performance is equivalent to the best reference models. It must be noted that this does not mean its performance is equivalent to other models since it was trained for a different data cohort. Therefore, its is expected that the imputed data model will have a slight performance drop when tested in a slightly different cohort. Table 7.3 shows that when only patients with missing BMI are taken into account the imputed version of the classifier neural network outperforms all models except the regression XBG, which is known to have high robustness to missing data [102]. Since we wish the model to reflect the entire cohort of patients, regardless of the availability of BMI data, the single point estimates suggest that the best

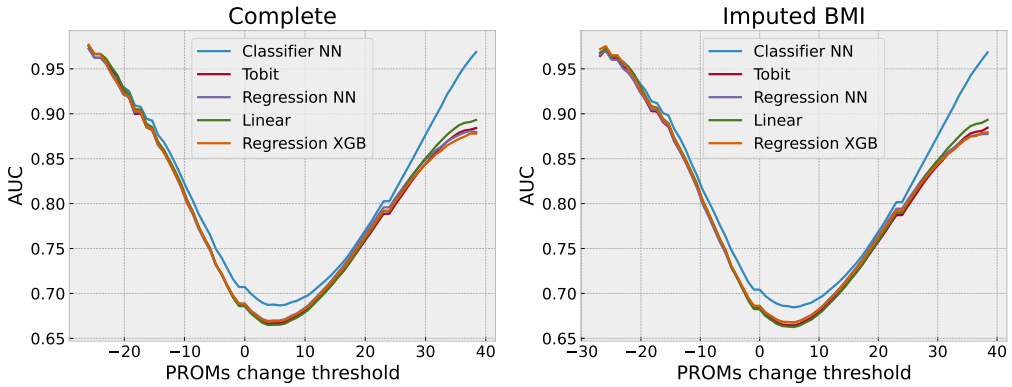


Figure 7.8.: AUC for the capability of the model to measure the probability of the change in PROMs score after a knee replacement surgery to be above a certain value. For each value in the x axis, it is possible to define a threshold and estimate the probability of the OKS variation to exceed this value. The figure gives the resulting AUC (y axis) for the probability of the OKS change to exceed each possible threshold (x axis). Panel (a) shows the results for the complete data cohort and panel (b) shows the results for the missing BMI cohort.

performing models are the imputed classifier neural network and the regression XGB.

The evaluation of the entire probability distribution is performed through Figures 7.7 and 7.8. Panel (a) of Figures 7.7 and 7.8 show significantly better results for both versions of the proposed model than other models. Panel (b) for the same figures show a further improvement for the imputed version of the model, which is consistent with the model including the missing BMI cohort in its training set.

The better performance in Figures 7.7 and 7.8 for the proposed model is a clear confirmation that the uncertainty of the outcome depends on the input variables as hypothesized in the beginning of this chapter. Additionally, the improved performance of the imputed version of the model in the missing BMI cohort confirms that the data is not missing completely at random (MCAR) and the imputed version of the model is the most appropriate, since restricting the training data to the complete data cohort would introduce estimation biases.

7.5 Relationship between input attributes and outcome

With the model validation complete, it is possible to apply the model to improve the understanding on how the specific attributes present in each surgery

affect the expected surgery outcome. For that purpose, we choose reference values for these attributes based on the frequency with which they appear in the dataset and then show how the expected outcome would vary for that reference patient if only one attributes changed. The reference value was chosen to be the median for numerical inputs and the mode for unordered categorical inputs. The reference surgery date was assumed to be the most recent date present in the dataset (28 March 2018). Figures 7.9, 7.10, 7.11,7.12 show the results for the OHS after hip replacement, the OKS after knee replacement, the EQ5D index after hip replacement and the VAS after hip replacement respectively.

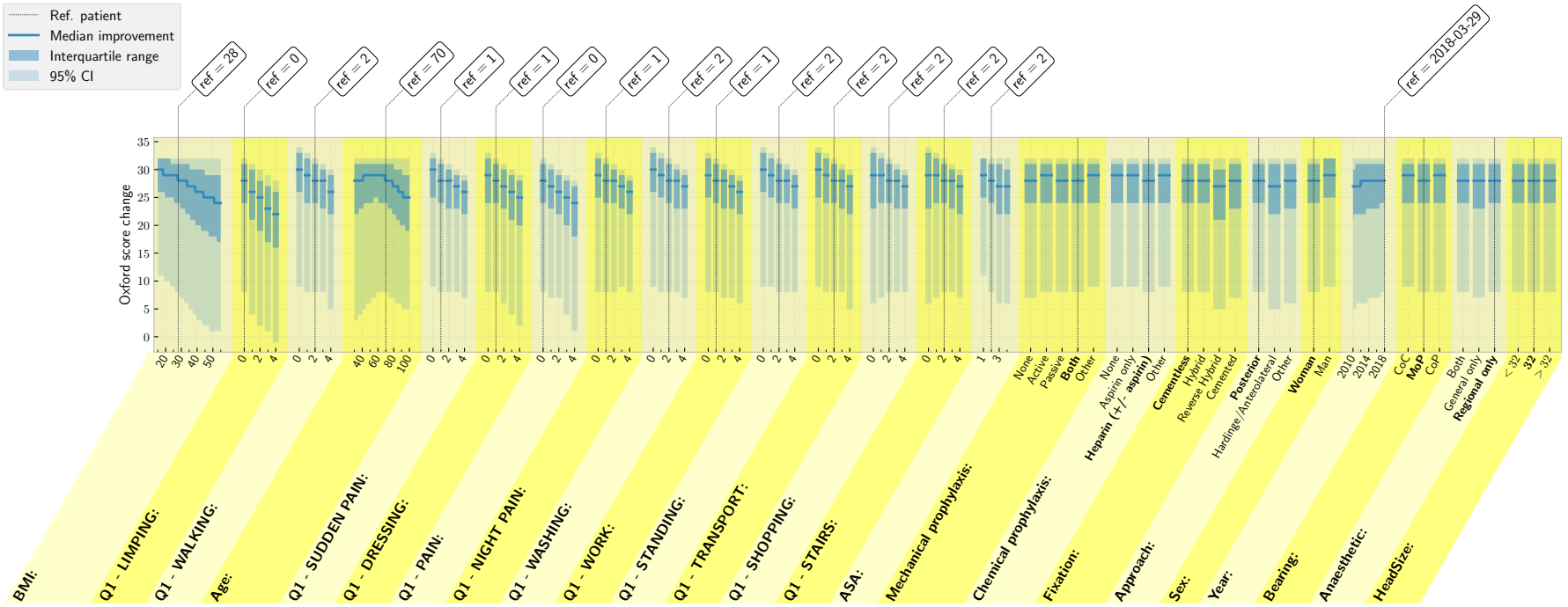


Figure 7.9.: Summary of the effect of each input variable to the postoperative OHS after a hip replacement. In each panel, one variable is changed and the others are kept at the reference value indicated.

Figure 7.10.: Summary of the effect of each input variable to the postoperative OKS after a knee replacement. In each panel, one variable is changed and the others are kept at the reference value indicated.

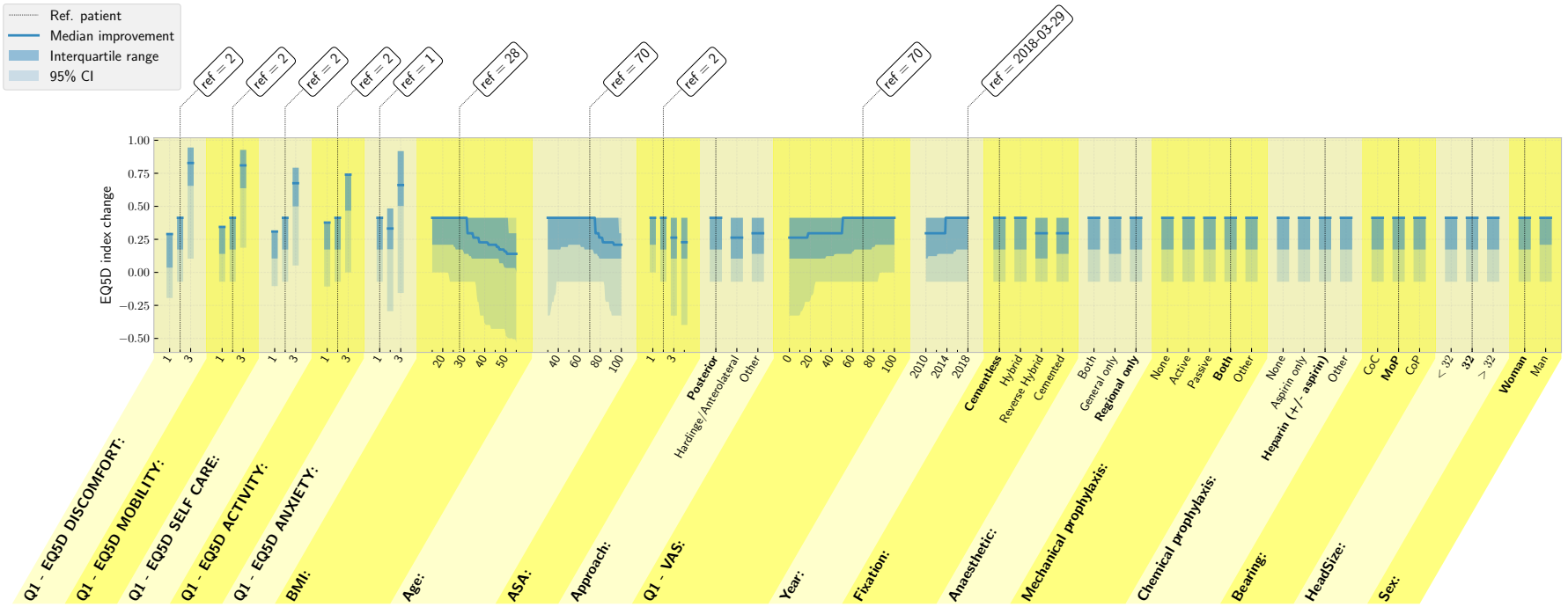


Figure 7.11.: Summary of the effect of each input variable to the postoperative EQ5D index after a hip replacement. In each panel, one variable is changed and the others are kept at the reference value indicated.

Figure 7.12.: Summary of the effect of each input variable to the postoperative VAS after a hip replacement. In each panel, one variable is changed and the others are kept at the reference value indicated.

8

Conclusions and future work

8.1 Conclusions

The major aim of this work was the development of machine learning methods to allow the prediction of joint replacement outcomes. That aim has been successfully achieved for the two different types of output that were studied, the first type being the prediction of the risks of death and revision, and the second the prediction of the patient perception of their health outcomes after the surgery. In the first type of prediction, the outcome is clearly defined and consists in the occurrence or not of some particular events as a function of the time passed after the surgery. This type of outcome is in the domain of survival analysis. In the second type of prediction, there exists no method for obtaining a fully comprehensive numerical description of the target outcome. Nonetheless, there are questionnaires that capture important aspects of it in a standardized manner which are known as patient reported outcome measures (PROMs) and were adopted as the outcome of the models. The method proposed for the solution of both tasks have the following important properties:

1. does not make any restrictive "a priori" assumptions about the patterns to be detected;

2. allows interpretability of the results.

In the survival analysis task, the final method proposed was the nested PH-MNN model with the use of transfer learning to avoid overfit. This method is the result of a sequence of novel developments that started with the development of the MNN framework, which has a hierarchical structure that enables analytical operations to be performed over its inputs and enhances interpretability of the model results. Theorem 1 shows the universal approximation property for MNN models, which means that this class of models can represent any continuous function with arbitrarily good accuracy provided that enough parameters are used. In Chapter 4, the MNN framework was used for survival modeling, showing better results than any previous models specially in its proportional hazards versions (PH-MNN and nested PH-MNN). In Chapter 5, a novel maximum likelihood approach was used to non-parametrically derive a class of models where the supremum of the likelihood is greater or equal to the supremum of the likelihood among all possible models. This result was proven in Theorem 2. The PH-MNN and nested PH-MNN models are within this class of models and their universal approximation property guarantees that with enough parameters they can approximate arbitrarily well any continuous function within this class of models. Additionally, Theorem 3 shows that these models can be estimated using the partial likelihood instead of the profile likelihood without change to the asymptotic behavior. Finally, in Chapter 6 a transfer learning strategy was proposed, allowing training to be performed in two steps where in the first there is data from a large number of patients but with fewer input variables and in the second all input variables are available but for a smaller number of patients. With the proposed transfer learning strategy, both datasets are combined in the estimation of the nested PH-MNN model, allowing overfit to be significantly reduced in the model with larger number of input variables. The nested PH-MNN model allows better interpretability when compared to other models. This is illustrated in Sections 4.5.2 and 6.3.

In the PROMs estimation task, the contribution was the reformulation of the modeling task as a classification problem instead of a regression problem as it is usually treated. There, the particular structure of the questionnaires used to obtain the PROMs scores impose a limited number of possible outcomes. Consequently, the estimation problem can be perfectly cast as a classification task where each possible outcome class corresponds to a possible PROMs score. This formulation as a classification task is more comprehensive than other formulation that have previously been used in the literature, including regression [93]

and binary classification [94]. Indeed, the classifier neural network proposed can account for all possible outcomes of the PROMs scores being modeled and thus allows any other representation of the outcome to be derived from it as shown in Section 7.1.3. Despite the classifier neural network employed in this task not allowing a comprehensive visualization of the model outcome as in Section 4.5.2, a display strategy was proposed that allows visualization of the main trends in the model as show in Figures 4.11 and 4.12. This strategy consists in defining a reference value for all input attributes in the model and displaying the model outcomes as a function of the variation of each input attribute one at a time.

8.2 Future work

There are several possibilities for future works is the extension of the PH-MNN model (in both the standard and the nested versions) to other types of estimation problems.

One example is the use of multimodal data as input of survival models, including medical imaging, electrocardiograms and others. The high computational efficiency for both training and testing in the PH-MNN model allows the use of deep neural networks for image feature extraction as part of survival models. Additionally, the extension of the nested PH-MNN model could provide an alternative for mixing data from different modalities.

The PH-MNN model would also be valuable for performing multi-task learning with time-to-event as one of the outputs. The profile likelihood proposed in Chapter 5 allows direct integration with the maximum likelihood estimation of additional outcomes. Form example, the survival estimates and PROMs estimates in the present work could be combined into a single neural network as an alternative or complement to the transfer learning strategy for avoiding overfit.

Another possible extension of the PH-MNN model would be in the modeling of electronic health records or other types of data that is distributed longitudinally in time, where the time between records can be modeled in the form of time-to-event and the records can be estimated jointly though maximum likelihood estimation.

Finally, the MNN framework is not restricted to survival modeling and could be applied to other machine learning tasks. One scenario were this could be beneficial is in other types of medical application were it is important to obtain an explicit interpretation of how the input variables affect the outcome. For example, in the postoperative PROMs estimation, an MNN model could be used in the form of a hierarchical model where a limited number of modes would describe the influence of age and BMI to the outcome and the other variables would provide the weights of each mode.

Another example is in multimodal learning, where the mixture model MNN structure could be used combine features coming from different data modalities outside the realm of survival analysis. This could be achieved by adapting the structure in Figure 3.3 by making $\mathbf{x}$ and $\mathbf{y}$ represent different modalities of

data where each modality could have their features extracted using a different neural network structure.

References

[1] Joseph G Ibrahim, Ming-Hui Chen and Debajyoti Sinha. *Bayesian survival analysis*. Springer Science & Business Media, 2001 (cit. on p. 10).

[2] David R Cox. `Regression models and life-tables'. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 34.2 (1972), pp. 187–202 (cit. on pp. 10, 16, 19, 33, 56, 65, 69).

[3] Ross L Prentice, John D Kalbfleisch, Arthur V Peterson Jr et al. `The analysis of failure times in the presence of competing risks'. In: *Biometrics* (1978), pp. 541–554 (cit. on pp. 11, 17, 58, 65, 69, 89, 101).

[4] Jason P Fine and Robert J Gray. `A proportional hazards model for the subdistribution of a competing risk'. In: *Journal of the American Statistical Association* 94.446 (1999), pp. 496–509 (cit. on pp. 11, 17, 33, 58, 65, 89, 101).

[5] RG Miller Jr, G Gong and A Munoz. *Survival analysis.* New York NY John Wiley 1981., 1981 (cit. on p. 12).

[6] Edward L Kaplan and Paul Meier. `Nonparametric estimation from incomplete observations'. In: *Journal of the American Statistical Association* 53.282 (1958), pp. 457–481 (cit. on pp. 15, 74, 92, 100).

[7] David G Hoel. `A representation of mortality data by competing risks'. In: *Biometrics* (1972), pp. 475–488 (cit. on pp. 15, 101).

[8] Wayne Nelson. `Theory and applications of hazard plotting for censored failure data'. In: *Technometrics* 14.4 (1972), pp. 945–966 (cit. on p. 15).

[9] Odd O Aalen and Søren Johansen. `An empirical transition matrix for non-homogeneous Markov chains based on censored observations'. In: *Scandinavian Journal of Statistics* (1978), pp. 141–150 (cit. on p. 16).

[10] Norman Breslow. `Covariance analysis of censored survival data'. In: *Biometrics* (1974), pp. 89–99 (cit. on pp. 17, 18, 66, 88).

[11] JD Kalbfleisch and Ross L Prentice. *Statistical analysis of failure time data*. Wiley, 1980 (cit. on pp. 17, 18, 66, 69, 88, 89, 96, 101).

[12] David Faraggi and Richard Simon. `A neural network model for survival data'. In: *Statistics in Medicine* 14.1 (1995), pp. 73–82 (cit. on p. 17).

[13] Jared L Katzman, Uri Shaham, Alexander Cloninger et al. `DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network'. In: *BMC Medical Research Methodology* 18.1 (2018), p. 24 (cit. on pp. 17, 69).

[14] Xinliang Zhu, Jiawen Yao and Junzhou Huang. `Deep convolutional neural network for survival analysis with pathological images'. In: *IEEE International Conference on Bioinformatics and Biomedicine*. IEEE. 2016, pp. 544–547 (cit. on p. 17).

[15] Margaux Luck, Tristan Sylvain, Héloıse Cardinal, Andrea Lodi and Yoshua Bengio. `Deep learning for patient-specific kidney graft survival analysis'. In: *arXiv preprint arXiv:1705.10245* (2017) (cit. on pp. 18, 19).

[16] Håvard Kvamme, Ørnulf Borgan and Ida Scheel. `Time-to-event prediction with neural networks and Cox regression'. In: *arXiv preprint arXiv:1907.00825* (2019) (cit. on pp. 18, 36, 57, 58, 69).

[17] Harald Heinzl, Alexandra Kaider and Gerhard Zlabinger. `Assessing interactions of binary time-dependent covariates with time in Cox proportional hazards regression models using cubic spline functions'. In: *Statistics in Medicine* 15.23 (1996), pp. 2589–2601 (cit. on p. 18).

[18] T Moreau, J O'quigley and M Mesbah. `A global goodness-of-fit statistic for the proportional hazards model'. In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 34.3 (1985), pp. 212–218 (cit. on p. 18).

[19] Robert J Gray. `Flexible methods for analyzing survival data using splines, with applications to breast cancer prognosis'. In: *Journal of the American Statistical Association* 87.420 (1992), pp. 942–951 (cit. on p. 18).

[20] Trevor Hastie and Robert Tibshirani. `Varying-coefficient models'. In: *Journal of the Royal Statistical Society: Series B* (*Methodological*) 55.4 (1993), pp. 757–779 (cit. on p. 18).

[21] Willi Sauerbrei, Carolina Meier-Hirmer, A Benner and Patrick Royston. `Multivariable regression model building by using fractional polynomials: description of SAS, STATA and R programs'. In: *Computational Statistics & Data Analysis* 50.12 (2006), pp. 3464–3485 (cit. on p. 18).

[22] Patrick Royston and Mahesh KB Parmar. `Flexible parametric proportional-hazards and proportional-odds models for censored survival data, with application to prognostic modelling and estimation of treatment effects'. In: *Statistics in Medicine* 21.15 (2002), pp. 2175–2197 (cit. on pp. 19, 49, 50, 88).

[23] PC Lambert, PW Dickman, CP Nelson and P Royston. `Estimating the crude probability of death due to cancer and other causes using relative survival models'. In: *Statistics in Medicine* 29.7-8 (2010), pp. 885–895 (cit. on pp. 19, 88).

[24] Rupert G Miller. `Least squares regression with censored data'. In: *Biometrika* 63.3 (1976), pp. 449–464 (cit. on pp. 19, 20).

[25] Jonathan Buckley and Ian James. `Linear regression with censored data'. In: *Biometrika* 66.3 (1979), pp. 429–436 (cit. on p. 20).

[26] Limin Peng and Yijian Huang. `Survival analysis with quantile regression models'. In: *Journal of the American Statistical Association* 103.482 (2008), pp. 637–649 (cit. on pp. 20, 69).

[27] Paidamoyo Chapfuwa, Chenyang Tao, Chunyuan Li et al. `Adversarial time-to-event modeling'. In: *arXiv preprint arXiv:1804.03184* (2018) (cit. on pp. 20, 30, 58).

[28] Pannagadatta K Shivaswamy, Wei Chu and Martin Jansche. `A support vector approach to censored targets'. In: *IEEE International Conference on Data Mining*. IEEE. 2007, pp. 655–660 (cit. on p. 20).

[29] Odd Aalen. `A model for nonparametric regression analysis of counting processes'. In: *Mathematical statistics and probability theory*. Springer, 1980, pp. 1–25 (cit. on pp. 21, 25).

[30] A Ciampi and J Etezadi-Amoli. `A general model for testing the proportional hazards and the accelerated failure time hypotheses in the analysis of censored survival data with covariates'. In: *Communications in Statistics-Theory and Methods* 14.3 (1985), pp. 651–667 (cit. on p. 22).

[31] Qixian Zhong, Jonas W Mueller and Jane-Ling Wang. `Deep extended hazard models for survival analysis'. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 15111–15124 (cit. on p. 22).

[32] Harald Steck, Balaji Krishnapuram, Cary Dehing-oberije, Philippe Lambin and Vikas C Raykar. `On ranking in survival analysis: bounds on the concordance index'. In: *Advances in Neural Information Processing Systems*. 2008, pp. 1209–1216 (cit. on p. 22).

[33] Frank E Harrell, Robert M Califf, David B Pryor, Kerry L Lee and Robert A Rosati. `Evaluating the yield of medical tests'. In: *Journal of the American Medical Association* 247.18 (1982), pp. 2543–2546 (cit. on pp. 22, 31).

[34] Vanya Van Belle, Kristiaan Pelckmans, JAK Suykens and Sabine Van Huffel. `Support vector machines for survival analysis'. In: *International Conference on Computational Intelligence in Medicine and Healthcare*. 2007, pp. 1–8 (cit. on p. 22).

[35] Ludger Evers and Claudia-Martina Messow. `Sparse kernel methods for high-dimensional survival data'. In: *Bioinformatics* 24.14 (2008), pp. 1632–1638 (cit. on p. 22).

[36] Anand Avati, Tony Duan, Kenneth Jung, Nigam H Shah and Andrew Ng. `Countdown regression: sharp and calibrated survival predictions'. In: *arXiv preprint arXiv:1806.08324* (2018) (cit. on p. 22).

[37] Rudolf Beran. *Nonparametric regression with randomly censored survival data*. University of California, Berkeley, 1981 (cit. on pp. 24, 25).

[38] Hemant Ishwaran and Min Lu. `Random survival forests'. In: *Wiley StatsRef: Statistics Reference Online* (2008), pp. 1–13 (cit. on p. 24).

[39] Hemant Ishwaran, Thomas A Gerds, Udaya B Kogalur et al. `Random survival forests for competing risks'. In: *Biostatistics* 15.4 (2014), pp. 757–773 (cit. on p. 25).

[40] Alexis Bellot and Mihaela van der Schaar. `Multitask boosting for survival analysis with competing risks'. In: *Advances in Neural Information Processing Systems*. 2018, pp. 1390–1399 (cit. on p. 25).

[41] George Chen. `Nearest neighbor and kernel survival analysis: nonasymptotic error bounds and strong consistency rates'. In: *International Conference on Machine Learning*. 2019, pp. 1001–1010 (cit. on p. 25).

[42] Chun-Nam Yu, Russell Greiner, Hsiu-Chin Lin and Vickie Baracos. `Learning patient-specific cancer survival distributions as a sequence of dependent regressors'. In: *Advances in Neural Information Processing Systems*. 2011, pp. 1845–1853 (cit. on p. 26).

[43] Stephane Fotso. `Deep neural networks for survival analysis based on a multi-task framework'. In: *arXiv preprint arXiv:1801.05512* (2018) (cit. on p. 26).

[44] Yan Li, Jie Wang, Jieping Ye and Chandan K Reddy. `A multi-task learning formulation for survival analysis'. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2016, pp. 1715–1724 (cit. on p. 26).

[45] Changhee Lee, William R Zame, Jinsung Yoon and Mihaela van der Schaar. `Deephit: a deep learning approach to survival analysis with competing risks'. In: *Association for the Advancement of Artificial Intelligence Conference*. 2018 (cit. on pp. 26, 36, 62, 69).

[46] Kan Ren, Jiarui Qin, Lei Zheng et al. `Deep recurrent survival analysis'. In: *Association for the Advancement of Artificial Intelligence Conference*. 2019, pp. 1–8 (cit. on p. 26).

[47] Eleonora Giunchiglia, Anton Nemchenko and Mihaela van der Schaar. `RNN-SURV: a deep recurrent model for survival analysis'. In: *International Conference on Artificial Neural Networks*. Springer. 2018, pp. 23–32 (cit. on p. 26).

[48] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart and Jimeng Sun. `Doctor AI: predicting clinical events via recurrent neural networks'. In: *Machine Learning for Healthcare Conference*. 2016, pp. 301–318 (cit. on p. 26).

[49] John D Kalbfleisch. `Non-parametric Bayesian analysis of survival time data'. In: *Journal of the Royal Statistical Society: Series B* (*Methodological*) 40.2 (1978), pp. 214–221 (cit. on p. 27).

[50] Sara Martino, Rupali Akerkar and Håvard Rue. `Approximate Bayesian inference for survival models'. In: *Scandinavian Journal of Statistics* 38.3 (2011), pp. 514–528 (cit. on p. 27).

[51] Heikki Joensuu, Aki Vehtari, Jaakko Riihimäki et al. `Risk of recurrence of gastrointestinal stromal tumour after surgery: an analysis of pooled population-based cohorts'. In: *The Lancet Oncology* 13.3 (2012), pp. 265–274 (cit. on p. 27).

[52] James E Barrett and Anthony CC Coolen. `Gaussian process regression for survival data with competing risks'. In: *arXiv preprint arXiv:1312.1591* (2013) (cit. on p. 27).

[53] Alan D Saul. `Gaussian process based approaches for survival analysis'. PhD thesis. University of Sheffield, 2016 (cit. on p. 27).

[54] Tamara Fernández, Nicolás Rivera and Yee Whye Teh. `Gaussian processes for survival analysis'. In: *Advances in Neural Information Processing Systems*. 2016, pp. 5021–5029 (cit. on p. 27).

[55] Ahmed M Alaa and Mihaela van der Schaar. `Deep multi-task Gaussian processes for survival analysis with competing risks'. In: *Advances in Neural Information Processing Systems*. 2017, pp. 2326–2334 (cit. on p. 28).

[56] Rajesh Ranganath, Linpeng Tang, Laurent Charlin and David Blei. `Deep exponential families'. In: *Artificial Intelligence and Statistics*. 2015, pp. 762–771 (cit. on p. 28).

[57] Rajesh Ranganath, Sean Gerrish and David Blei. `Black box variational inference'. In: *Artificial Intelligence and Statistics*. 2014, pp. 814–822 (cit. on p. 28).

[58] Rajesh Ranganath, Adler Perotte, Noémie Elhadad and David Blei. `Deep survival analysis'. In: *arXiv preprint arXiv:1608.02158* (2016) (cit. on p. 29).

[59] Xenia Miscouridou, Adler Perotte, Noémie Elhadad and Rajesh Ranganath. `Deep survival analysis: nonparametrics and missingness'. In: *Machine Learning for Healthcare Conference*. 2018, pp. 244–256 (cit. on p. 29).

[60] Quan Zhang and Mingyuan Zhou. `Nonparametric Bayesian Lomax delegate racing for survival analysis with competing risks'. In: *Advances in Neural Information Processing Systems*. 2018, pp. 5002–5013 (cit. on p. 29).

[61] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza et al. `Generative adversarial nets'. In: *Advances in Neural Information Processing Systems*. 2014, pp. 2672–2680 (cit. on p. 30).

[62] Xinhua Liu and Zhezhen Jin. `A non-parametric approach to scale reduction for uni-dimensional screening scales'. In: *The International Journal of Biostatistics* 5.1 (2009) (cit. on p. 31).

[63] Thomas A Gerds, Michael W Kattan, Martin Schumacher and Changhong Yu. `Estimating a time-dependent concordance index for survival prediction models with covariate dependent censoring'. In: *Statistics in Medicine* 32.13 (2013), pp. 2173–2184 (cit. on p. 31).

[64] Marcel Wolbers, Paul Blanche, Michael T Koller, Jacqueline CM Witteman and Thomas A Gerds. `Concordance for prognostic models with competing risks'. In: *Biostatistics* 15.3 (2014), pp. 526–539 (cit. on p. 32).

[65] Laura Antolini, Patrizia Boracchi and Elia Biganzoli. `A time-dependent discrimination index for survival data'. In: *Statistics in Medicine* 24.24 (2005), pp. 3927–3944 (cit. on p. 32).

[66] Glenn W Brier. `Verification of forecasts expressed in terms of probability'. In: *Monthly weather review* 78.1 (1950), pp. 1–3 (cit. on p. 32).

[67] Edward L Korn and Richard Simon. `Measures of explained variation for survival data'. In: *Statistics in Medicine* 9.5 (1990), pp. 487–503 (cit. on p. 32).

[68] Erika Graf, Claudia Schmoor, Willi Sauerbrei and Martin Schumacher. `Assessment and comparison of prognostic classification schemes for survival data'. In: *Statistics in Medicine* 18.17-18 (1999), pp. 2529–2545 (cit. on p. 32).

[69] Thomas A Gerds and Martin Schumacher. `Consistent estimation of the expected Brier score in general survival models with right-censored event times'. In: *Biometrical Journal* 48.6 (2006), pp. 1029–1040 (cit. on p. 33).

[70] Fabio Luis de Mello, J Mark Wilkinson and Visakan Kadirkamanathan. `Metaparametric Neural Networks for Survival Analysis'. In: *IEEE Transactions on Neural Networks and Learning Systems* (2021) (cit. on pp. 38, 57, 68, 100).

[71] Richard B Holmes. *A course on optimization and best approximation*. Vol. 257. Springer, 1972 (cit. on p. 42).

[72] Brook Taylor. *Methodus incrementorum directa et inversa*. Innys, 1717 (cit. on p. 47).

[73] Robert A Jacobs, Michael I Jordan, Steven J Nowlan and Geoffrey E Hinton. `Adaptive mixtures of local experts'. In: *Neural computation* 3.1 (1991), pp. 79–87 (cit. on p. 54).

[74] Michael I Jordan and Robert A Jacobs. `Hierarchical mixtures of experts and the EM algorithm'. In: *Neural computation* 6.2 (1994), pp. 181–214 (cit. on p. 54).

[75] Christopher M Bishop. `Mixture density networks'. In: (1994) (cit. on p. 55).

[76] Frédéric Riesz. `Sur la convergence en moyenne'. In: *Acta Sci. Math* 4.1 (1928), pp. 58–64 (cit. on p. 62).

[77] Martin G Larson and Gregg E Dinse. `A mixture model for the regression analysis of competing risks data'. In: *Journal of the Royal Statistical Society: Series C* (*Applied Statistics*) 34.3 (1985), pp. 201–211 (cit. on pp. 65, 89).

[78] D Randall Wilson and Tony R Martinez. `The general inefficiency of batch training for gradient descent learning'. In: *Neural networks* 16.10 (2003), pp. 1429–1451 (cit. on p. 65).

[79] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever and Ruslan Salakhutdinov. `Dropout: a simple way to prevent neural networks from overfitting'. In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958 (cit. on p. 68).

[80] Sebastian Pölsterl. `scikit-survival: A Library for Time-to-Event Analysis Built on Top of scikit-learn.' In: *J. Mach. Learn. Res.* 21.212 (2020), pp. 1–6 (cit. on p. 70).

[81] Anastasios A Tsiatis et al. `A large sample study of Cox's regression model'. In: *The Annals of Statistics* 9.1 (1981), pp. 93–108 (cit. on p. 88).

[82] Per Kragh Andersen and Richard D Gill. `Cox's regression model for counting processes: a large sample study'. In: *The annals of statistics* (1982), pp. 1100–1120 (cit. on p. 88).

[83] Søren Johansen. `An extension of Cox's regression model'. In: *International Statistical Review/Revue Internationale de Statistique* (1983), pp. 165–174 (cit. on p. 89).

[84] Judy P Sy and Jeremy MG Taylor. `Estimation in a Cox proportional hazards cure model'. In: *Biometrics* 56.1 (2000), pp. 227–236 (cit. on pp. 89, 96).

[85] SK Ng and GJ McLachlan. `An EM-based semi-parametric mixture model approach to the regression analysis of competing-risks data'. In: *Statistics in Medicine* 22.7 (2003), pp. 1097–1111 (cit. on p. 89).

[86] I-Shou Chang, Chao A Hsiung, Chi-Chung Wen, Yuh-Jenn Wu and Che-Chi Yang. `Non-parametric maximum-likelihood estimation in a semi-parametric mixture model for competing-risks data'. In: *Scandinavian Journal of Statistics* 34.4 (2007), pp. 870–895 (cit. on p. 89).

[87] Nir Friedman and Zohar Yakhini. `On the sample complexity of learning Bayesian networks'. In: *arXiv preprint arXiv:1302.3579* (2013) (cit. on p. 106).

[88] Thomas R Fleming and David P Harrington. *Counting processes and survival analysis*. John Wiley & Sons, 2011 (cit. on p. 106).

[89] Jill Dawson, Ray Fitzpatrick, Andrew Carr and David Murray. `Questionnaire on the perceptions of patients about total hip replacement'. In: *The Journal of bone and joint surgery. British volume* 78.2 (1996), pp. 185–190 (cit. on pp. 114, 131).

[90] Jill Dawson, Ray Fitzpatrick, David Murray and Andrew Carr. `Questionnaire on the perceptions of patients about total knee replacement'. In: *The Journal of bone and joint surgery. British volume* 80.1 (1998), pp. 63–69 (cit. on pp. 114, 131).

[91] The EuroQol Group. `EuroQol-a new facility for the measurement of health-related quality of life'. In: *Health policy* 16.3 (1990), pp. 199–208 (cit. on p. 131).

[92] Adrian Sayers, Michael R Whitehouse, Andrew Judge et al. `Analysis of change in patient-reported outcome measures with floor and ceiling effects using the multilevel Tobit model: a simulation study and an example from a National Joint Register using body mass index and the Oxford Hip Score'. In: *BMJ open* 10.8 (2020), e033646 (cit. on pp. 132, 144).

[93] Manuel Huber, Christoph Kurz and Reiner Leidl. `Predicting patient-reported outcomes following hip and knee replacement surgery using supervised machine learning'. In: *BMC medical informatics and decision making* 19.1 (2019), pp. 1–13 (cit. on pp. 132, 144, 154).

[94] Mark Alan Fontana, Stephen Lyman, Gourab K Sarker, Douglas E Padgett and Catherine H MacLean. 'Can machine learning algorithms predict which patients will achieve minimally clinically important differences from total joint arthroplasty?' In: *Clinical orthopaedics and related research* 477.6 (2019), p. 1267 (cit. on pp. 132, 155).

[95] Jos Twisk and Frank Rijmen. 'Longitudinal tobit regression: a new approach to analyze outcome variables with floor or ceiling effects'. In: *Journal of clinical epidemiology* 62.9 (2009), pp. 953–958 (cit. on p. 134).

[96] Andy Liaw, Matthew Wiener et al. 'Classification and regression by randomForest'. In: *R news* 2.3 (2002), pp. 18–22 (cit. on p. 134).

[97] Aleksi Reito, Anni Järvistö, Esa Jämsen et al. 'Translation and validation of the 12-item Oxford knee score for use in Finland'. In: *BMC Musculoskeletal Disorders* 18.1 (2017), pp. 1–6 (cit. on p. 135).

[98] A Paulsen, Anders Odgaard and S Overgaard. 'Translation, cross-cultural adaptation and validation of the Danish version of the Oxford hip score: assessed against generic and disease-specific questionnaires'. In: *Bone & joint research* 1.9 (2012), pp. 225–233 (cit. on p. 135).

[99] John S Bridle. 'Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition'. In: *Neurocomputing*. Springer, 1990, pp. 227–236 (cit. on p. 136).

[100] Diederik P Kingma and Jimmy Ba. 'Adam: A method for stochastic optimization'. In: *arXiv preprint arXiv:1412.6980* (2014) (cit. on p. 136).

[101] Donald B Rubin. 'Characterizing the estimation of parameters in incomplete-data problems'. In: *Journal of the American Statistical Association* 69.346 (1974), pp. 467–474 (cit. on p. 139).

[102] Pasha Khosravi, Antonio Vergari, YooJung Choi, Yitao Liang and Guy Van den Broeck. 'Handling missing data in decision trees: A probabilistic approach'. In: *arXiv preprint arXiv:2006.16341* (2020) (cit. on p. 145).

A

Survival model pseudo code

In this appendix we provide the pseudo-codes for the survival models implemented in this thesis.

Algorithm 5 Computation of the hazard ratio (ω_j) and log-likelihood (ℓ) in the PH-MNN model.

Require: dataset $\mathcal{D}$
Require: $n_b, n_e > 0$
  $N \leftarrow$ number of subjects in $\mathcal{D}$
  for j in event types **do**
    $N_j \leftarrow$ number of instances of event j in $\mathcal{D}$
    for $\boldsymbol{x}_n$ in $\mathcal{D}$ **do**
      $\omega_{n,j} \leftarrow \sum_k \exp(\psi_{k,j}(\boldsymbol{x}_n))\nu_k(t)$
    end for
    $\boldsymbol{\omega}_{b,j}, \boldsymbol{T}_b \leftarrow n_b$ random samples from $\mathcal{D}$
    $\boldsymbol{\omega}_{e,j}, \boldsymbol{T}_{e,j} \leftarrow n_e$ random samples from $\mathcal{D}$ so that $E_n = 1$ and $j_n = j$
    $\ell_{e,j} \leftarrow \log \omega_{e,j} - \log \sum_{b|t_b \geq t_e} \omega_{b,j} - \log N/n_b$
    $\ell_j \leftarrow$ sum of $\ell_{e,j}$ for all event samples of type j.
  end for
  $\ell \leftarrow \sum_j [\ell_j N_j / n_e]$

Algorithm 6 Computation of the cause specific cummulative hazard function for a given subject n ($\Lambda_{n,j}$) and log-likelihood (ℓ_n) in the QR-MNN model.

Require: dataset $\mathcal{D}$
Require: quantile knots $\exp(-\Lambda_k)$ with $\Lambda_k \in \{\Lambda_1, \ldots, \Lambda_K\}$
  for j in event types **do**
    $\Lambda_0 \leftarrow 0$
    $T_{n,j,0} \leftarrow 0$
    for Λ_k in $\{\Lambda_1, \ldots, \Lambda_K\}$ **do**
      $T_{n,j,k} \leftarrow T_{n,j,k-1} + \exp(\psi_{k,j}(\boldsymbol{x}_n))$
    end for
    $\Lambda_{n,j}(t) \leftarrow$ linear interpolation of Λ_k with x-axis knots given by $T_{n,j,k}$.
    $\lambda_{n,j}(t) \leftarrow$ time derivative of $\Lambda_{n,j}(t)$
  end for
  $\ell_n = E_n \log(\lambda_{n,j_n}(T_n)) - \sum_j \Lambda_{n,j}(T_n)$

Algorithm 7 Computation of the cause specific cummulative hazard function for a given subject n ($\Lambda_{n,j}$) and log-likelihood (ℓ_n) in the DR-MNN model.

Require: dataset $\mathcal{D}$
Require: time knots $t_k \in \{t_1, \ldots, t_K\}$
  for j in event types **do**
    $T_0 \leftarrow 0$
    $\Lambda_{n,j,0} \leftarrow 0$
    for t_k in $\{t_1, \ldots, t_K\}$ **do**
      $\Lambda_{n,j,k} \leftarrow \Lambda_{n,j,k-1} + \exp(\psi_{k,j}(\boldsymbol{x}_n))$
    end for
    $\Lambda_{n,j}(t) \leftarrow$ linear interpolation of $\Lambda_{n,j,k}$ with x-axis knots given by t_k.
    $\lambda_{n,j}(t) \leftarrow$ time derivative of $\Lambda_{n,j}(t)$
  end for
  $\ell_n = E_n \log(\lambda_{n,j_n}(T_n)) - \sum_j \Lambda_{n,j}(T_n)$

Algorithm 8 Computation of the hazard ratio (ω_j) and log-likelihood (ℓ) in the Cox model.

Require: dataset $\mathcal{D}$
Require: $n_b, n_e > 0$
  $N \leftarrow$ number of subjects in $\mathcal{D}$
  for j in event types **do**
    $N_j \leftarrow$ number of instances of event j in $\mathcal{D}$
    for $\boldsymbol{x}_n$ in $\mathcal{D}$ **do**
      $\omega_{n,j} \leftarrow \exp(\boldsymbol{\beta}_j^T \boldsymbol{x}_n)$
    end for
    $\boldsymbol{\omega}_{b,j}, \boldsymbol{T}_b \leftarrow n_b$ random samples from $\mathcal{D}$
    $\boldsymbol{\omega}_{e,j}, \boldsymbol{T}_{e,j} \leftarrow n_e$ random samples from $\mathcal{D}$ so that $E_n = 1$ and $j_n = j$
    $\ell_{e,j} \leftarrow \log \omega_{e,j} - \log \sum_{b|t_b \geq t_e} \omega_{b,j} - \log N/n_b$
    $\ell_j \leftarrow$ sum of $\ell_{e,j}$ for all event samples of type j.
  end for
  $\ell \leftarrow \sum_j [\ell_j N_j / n_e]$

Algorithm 9 Computation of the cause specific cummulative hazard function for a given subject n ($\Lambda_{n,j}$) and training cost (ϵ) in the quantile regression model.

Require: dataset $\mathcal{D}$
Require: quantile knots $\tau_k \in \{\tau_1, \ldots, \tau_K\}$
  $G(t) \leftarrow$ Kaplan-Meier estimation of the probability of censoring not happening until t
  for j in event types **do**
    for k in $\{1, \ldots, K\}$ **do**
      $Q_{n,j,k} \leftarrow \exp(\boldsymbol{\beta}_{j,k}^T \boldsymbol{x}_n)$
    end for
    $\Lambda_{n,j}(t) \leftarrow$ linear interpolation of $-\log \tau_k$ with x-axis knots given by $Q_{n,j,k}$
    $\boldsymbol{x}_{b,j}, \boldsymbol{T}_{e,j} \leftarrow n_b$ random samples from $\mathcal{D}$
    $\boldsymbol{x}_{e,j}, \boldsymbol{T}_{e,j} \leftarrow n_e$ random samples from $\mathcal{D}$ so that $E_n = 1$ and $j_n = j$
    $e_j^{(1)} \leftarrow \sum_{k,e} |\log T_e - \log Q_{e,j,k}| N_j / (K n_e G(T_e) N)$
    $e_j^{(2)} \leftarrow |500 + \sum_{k,e} (\log Q_{e,j,k}) N_j / (K n_e G(T_e) N)|$
    $b_j \leftarrow |500 - \sum_{k,b} (2 \log Q_{b,j,k} \tau_k) / (K n_b)|$
  end for
  $\epsilon = \sum_j (e_j^{(1)} + e_j^{(2)} + b_j)$

Algorithm 10 Computation of the hazard ratio (ω_j) and log-likelihood (ℓ) in the DeepSurv model.

Require: dataset $\mathcal{D}$
Require: $n_b, n_e > 0$
  $N \leftarrow$ number of subjects in $\mathcal{D}$
  for j in event types **do**
    $N_j \leftarrow$ number of instances of event j in $\mathcal{D}$
    for $\boldsymbol{x}_n$ in $\mathcal{D}$ **do**
      $\omega_{n,j} \leftarrow \exp(\psi_j(\boldsymbol{x}_n))$
    end for
    $\boldsymbol{\omega}_{b,j}, \boldsymbol{T}_b \leftarrow n_b$ random samples from $\mathcal{D}$
    $\boldsymbol{\omega}_{e,j}, \boldsymbol{T}_{e,j} \leftarrow n_e$ random samples from $\mathcal{D}$ so that $E_n = 1$ and $j_n = j$
    $\ell_{e,j} \leftarrow \log \omega_{e,j} - \log \sum_{b|t_b \geq t_e} \omega_{b,j} - \log N/n_b$
    $\ell_j \leftarrow$ sum of $\ell_{e,j}$ for all event samples of type j.
  end for
  $\ell \leftarrow \sum_j [\ell_j N_j / n_e]$

Algorithm 11 Computation of the hazard ratio (ω_j) and log-likelihood (ℓ) in the Cox-Time model.

Require: dataset $\mathcal{D}$
Require: $n_b, n_e > 0$
  $N \leftarrow$ number of subjects in $\mathcal{D}$
  for j in event types **do**
    $N_j \leftarrow$ number of instances of event j in $\mathcal{D}$
    for $\boldsymbol{x}_n$ in $\mathcal{D}$ **do**
      $\omega_{n,j}(T_n) \leftarrow \exp(\psi_j(\boldsymbol{x}_n, T_n))$
    end for
    $\boldsymbol{\omega}_{b,j}, \boldsymbol{T}_b \leftarrow n_b$ random samples from $\mathcal{D}$
    $\boldsymbol{\omega}_{e,j}, \boldsymbol{T}_{e,j} \leftarrow n_e$ random samples from $\mathcal{D}$ so that $E_n = 1$ and $j_n = j$
    for b in samples $\boldsymbol{\omega}_{b,j}, \boldsymbol{T}_b$ **do**
      for e in event samples $\boldsymbol{\omega}_{e,j}, \boldsymbol{T}_{e,j}$ **do**
        $\omega_{b,j}(T_e) \leftarrow \exp(\psi_j(\boldsymbol{x}_b, T_e))$
      end for
    end for
    $\ell_{e,j} \leftarrow \log \omega_{e,j}(T_e) - \log \sum_{b|t_b \geq t_e} \omega_{b,j}(T_e) - \log N/n_b$
    $\ell_j \leftarrow$ sum of $\ell_{e,j}$ for all event samples of type j.
  end for
  $\ell \leftarrow \sum_j [\ell_j N_j / n_e]$

Algorithm 12 Computation of cummulative incidence function (F_j) and training loss (ϵ) in the DeepHit model.

Require: dataset $\mathcal{D}$
Require: time knots $t_k \in \{t_1, \ldots, t_K\}$
Require: $n_b, n_e > 0$
  $N \leftarrow$ number of subjects in $\mathcal{D}$
  for j in event types **do**
    $N_j \leftarrow$ number of instances of event j in $\mathcal{D}$
    $T_0 \leftarrow 0$
    $F_{n,j} \leftarrow 0$
    for t_k in $\{t_1, \ldots, t_K\}$ **do**
      $f_{n,j,k} \leftarrow \exp(\psi_{k,j}(\boldsymbol{x}_n)) / \sum_{k,j} \exp(\psi_{k,j}(\boldsymbol{x}_n))$
      if $T_n \geq t_k$ **then**
        $F_{n,j} \leftarrow F_{n,j} + f_{n,j,k}$
      end if
    end for
    $\boldsymbol{\omega}_{b,j}, \boldsymbol{T}_b \leftarrow n_b$ random samples from $\mathcal{D}$
    $\boldsymbol{\omega}_{e,j}, \boldsymbol{T}_{e,j} \leftarrow n_e$ random samples from $\mathcal{D}$ so that $E_n = 1$ and $j_n = j$
    $L_{2,e} \leftarrow 0$
    for e in event samples $\boldsymbol{\omega}_{e,j}, \boldsymbol{T}_{e,j}$ **do**
      for b in samples $\boldsymbol{\omega}_{b,j}, \boldsymbol{T}_b$ **do**
        $L_{2,e} \leftarrow L_{2,e} + \exp(-10(f_{e,j,k} - f_{b,j,k}))n_e/N$
      end for
      $k_n \leftarrow n | t_n \geq T_n; t_{n-1} < T_n$
      $\epsilon \leftarrow -\log[(N_j/N) f_{n,j,k_n} / (1 - F_{n,j})] + 0.1 L_{2,e}$
    end for
  end for
  $\epsilon \leftarrow \sum_e \epsilon + \sum_b \log(1 - F_{b,j})/n_b$