Integrating longitudinal data for enhanced survival analysis : methods and applications

van Tuan Nguyen

UNIVERSITÉ PARIS CITÉ

LPSM

École doctorale **École Doctorale Sciences Mathématiques de Paris Centre**

Unité de recherche **Laboratoire de Probabilités, Statistique et Modélisation**

Integrating longitudinal data for enhanced survival analysis : methods and applications

Par VAN TUAN NGUYEN

Thèse de doctorat de STATISTIQUES

Dirigée par AGATHE GUILLOUX

Présentée et soutenue publiquement le 12 Décembre 2024

Devant un jury composé de :

ROBIN GENUER, MCF-HDR	Université de Bordeaux	Rapporteur
OLIVIER BOUAZIZ, PROFESSEUR HDR	Université de Lille	Rapporteur
STEPHANE ROBIN, PROFESSEUR HDR	Sorbonne Université	Examinateur
AGATHE GUILLOUX, DR-HDR	INRIA Paris	Directrice de thèse
ADELINE FERMANIAN, PHD	Califrais	Membre invitée

This work was funded by the Califrais company via the CIFRE program.

"If we knew what we were doing, it wouldn't be called research."

Albert Einstein

"A journey of a thousand miles begins with a single step."

Lao Tzu

Acknowledgements

Time flies so fast, and everything unfolds within that time. It feels like just yesterday I was taking my first steps into the world of academic research, and now, here I am, sitting to write the final lines of my dissertation, looking back on the journey of over three years that has led me to this point. It has been a path full of ups and downs, with many changes and valuable lessons that will guide me forward. Reflecting on this journey, I have come to realize just how much help and support I have received. With this in mind, I would like to express my deep gratitude to everyone who has supported and guided me throughout this research journey. This thesis would not have been completed without your encouragement, expertise, and enthusiasm.

First and foremost, I am deeply thankful to my supervisors: Agathe, Adeline, and Simon. Your constant guidance, support, and patience have been crucial in completing this thesis, and I truly appreciate everything you have done for me.

Agathe, I cannot find the right words to express how much I appreciate your dedication and enthusiastic mentorship. Thank you for agreeing to be my supervisor during such a challenging time in my research. Your deep knowledge and detailed feedback allowed me to achieve results that went beyond my expectations. I am especially grateful for your patience, which helped me not only finish this thesis but also gain confidence and clarity in my work on survival analysis. I feel incredibly fortunate to have had the opportunity to work under your guidance.

Adeline, I am extremely grateful for all the help you have given me throughout my PhD. Your insightful advice, careful attention to detail, wealth of experience, and honest, helpful suggestions have guided my progress. Thank you for being there when I felt uncertain or stuck in this journey. Your encouragement, inspiration, and friendly attitude have been a source of motivation, helping me regain the confidence I needed to keep going.

Simon, I am very thankful for your support from the very beginning of my time at Califrais. Your enthusiastic help in connecting me with this PhD program, as well as your invaluable expertise in the early stages of my research, have been incredibly crucial.

I also want to extend my thanks to Stéphane, my initial supervisor, for the solid guidance you gave me in the early stages of this work. Your support and advice helped lay the foundation for the research that followed.

Next, I would like to express my sincere thanks to my reporters - Robin and Olivier - for taking the time to review my dissertation. Your insightful feedback has deepened my understanding of the topic and has been a huge help in guiding future steps for my research. I also want to express my appreciation to Stéphane for attending my defense jury. I feel honored by your presence.

Thank you, Linus, for collaborating with me on part of this thesis. We achieved great results together, and I will always remember the time we spent working at Sorbonne Uni-

versity, at SCAI, and during the conference in Vienna.

Thanks to all my colleagues at Califrais for their continuous support in both work and life. Thanks to Simon, Massil, Jean Philippe, Jean Luc, Sobian, Ali, and William for sharing interesting ideas during our weekly seminars. Thank you, Simon, for assisting with collecting customer ordering data for my research. Thanks to Massil for your warm and dedicated support during the preparation of my thesis defense.

I also want to thank the HeKa team for your support during the final phase of this dissertation.

Thank you to my dear friends - Quyen, Binh, Thuong, and Tuyen - for your unwavering support and care. Even our light-hearted chats helped relieve so much stress during the times I felt stuck and almost gave up.

I want to thank my sisters - Duyen, Mai, and Vien - for your help, encouragement, and shared moments. Thanks to my friends in Paris — Au, Kien, Truong, Ninh, Nhu, Yen, Tuan, and Tri. Thank you for always making me feel connected and bringing joy to my life.

Thank you, Hanh, my soul mate. Knowing you has been one of the greatest blessings of my life. Thank you for always listening to my struggles and offering me valuable advice. Together, we have worked through the ups and downs of our PhD journeys and, in the end, we both succeeded.

I want to express my deepest gratitude to my family, who have always been my strong support throughout this journey. Thank you, my siblings, for always believing in me and sharing both the joys and the sorrows along the way. A special thank you to my parents, who have always been my steady anchor. I know that, no matter how many failures or difficulties I face, you will always be there, encouraging me and helping me get back up. This truly means the world to me.

Finally, I want to thank everyone who has been part of this journey, whether in work or in life. Every bit of help and every word of encouragement has been an incredible source of motivation that has helped me complete this dissertation. Once again, I sincerely thank you all!

Résumé

Titre: Intégration des données longitudinales pour une analyse de survie améliorée: méthodes et applications

Résumé: La disponibilité croissante des données longitudinales offre des opportunités significatives pour améliorer les modèles de survie en permettant des mises à jour dynamiques des évaluations des risques au fil du temps. Cependant, l'intégration de ces données dans les modèles de survie reste limitée en raison de la complexité des données longitudinales, des défis computationnels posés par les ensembles de données de haute dimension, et des difficultés d'interprétation. Cette limitation crée un écart crucial dans le domaine, restreignant la capacité à faire des prédictions précises et en temps réel dans des situations où les facteurs de risque évoluent au fil du temps. Cette thèse vise à combler cet écart en développant de nouveaux cadres d'analyse de survie qui intègrent efficacement les données de survie censurées aux côtés des données longitudinales multivariées. Le premier cadre propose un nouveau modèle conjoint dont une caractéristique clé est l'utilisation de caractéristiques génériques extraites des données longitudinales directement dans le modèle de survie. En outre, ces caractéristiques génériques sont indépendantes des hypothèses du modèle longitudinal, ce qui les rend adaptées aux marqueurs longitudinaux de haute dimension. Le second cadre s'appuie sur les récentes avancées en apprentissage profond et en équations différentielles pour apprendre des états latents guidés par les données, qui sont ensuite utilisés pour modéliser la fonction d'intensité des processus de comptage. Ces méthodes proposées démontrent de solides performances prédictives dans des expériences d'analyse de survie en temps réel et sont conçues pour être à la fois évolutives pour les problèmes de haute dimension. L'application pratique de ces cadres est illustrée par leur utilisation dans la prédiction de l'attrition chez Califrais. En intégrant efficacement les données de commandes des clients, ces modèles fournissent des prédictions plus précises du risque d'attrition, permettant ainsi aux entreprises de prendre des mesures proactives pour fidéliser leurs clients. Cette recherche contribue au domaine en fournissant des outils avancés pour l'analyse de survie et établit une base pour de futurs développements visant à intégrer pleinement les données longitudinales dans les modèles de survie dans divers domaines appliqués.

Mots clefs: Analyse de survie; Données longitudinales; Équation différentielle contrôlée par réseau neuronal; Modèle conjoint; Prédiction de l'attrition; Signature; Statistiques à haute dimension.

Title: Integrating longitudinal data for enhanced survival analysis: methods and applications

Abstract: The increasing availability of longitudinal data offers significant opportunities to improve survival models by allowing dynamic updates to risk assessments over time. However, the integration of this data into survival models remains limited due to the complexity of longitudinal data, computational challenges posed by high-dimensional datasets, and difficulties in interpretation. This limitation creates a critical gap in the field, restricting the ability to make accurate and real-time predictions in situations where risk factors change over time. This thesis aims to bridge this gap by developing new frameworks for survival analysis that effectively incorporate censored survival data alongside multivariate longitudinal data. The first framework develops a new joint model where a key feature is the use of generic features extracted from the longitudinal data directly in the survival model. In addition, these generic features are independent of assumptions in the longitudinal model making it suitable for high-dimensional longitudinal markers. The second framework leverages recent advances in deep learning and differential equations to learn data-driven latent states, which are then used to model the intensity function of counting processes. These proposed methods demonstrate strong predictive performance in extensive real-time survival analysis experiments and are designed to be both user-friendly and scalable for high-dimensional problems. The practical application of these frameworks is illustrated through their use in churn prediction at Califrais. By effectively integrating customer order data, these models provide more accurate predictions of churn risk, allowing businesses to take proactive measures to retain customers. This research contributes to the field by providing advanced tools for survival analysis and establishes a foundation for further developments aimed at fully integrating longitudinal data into survival models in various applied domains.

Keywords: Churn Prediction; High-dimensional Statistics; Joint Model; Longitudinal Data; Neural Controlled Differential Equation; Signature; Survival Analysis.

Contents

Chapter 1

Introduction

Contents

1.1 Churn prediction at Califrais

This thesis, which has benefited from the CIFRE program, was conducted at the company Califrais and Université Paris Cité. We present in this introductory chapter the industrial context of this thesis.

Califrais

Califrais was created in July 2014 with the aim of creating a modern distribution service for fresh food from suppliers at the Rungis market, the largest fresh produce market in the world, to restaurants in Paris. As the global population rapidly grows, wholesale markets are becoming increasingly important in ensuring the world's food supply. However, many sectors face inefficiencies in their supply chains, such as reliance on paper catalogs, manual processes, and opaque pricing, leading to significant challenges in transactions between customers and suppliers. Customers spend considerable time sourcing, managing supply, and negotiating with multiple suppliers, while suppliers struggle to understand customer needs and optimize transportation. The result is significant food waste and increased CO2 emissions. The problem is even worse in the fresh food supply sector, where products are highly perishable. These inefficiencies were particularly evident at Rungis in the 2010s. Califrais' solution focuses on streamlining food distribution through the mutualization of

orders and logistics. By consolidating orders from multiple suppliers into a single, efficient delivery process, Califrais optimizes the supply chain, reduces waste, and minimizes environmental impact while improving the overall efficiency of fresh food distribution for both suppliers and customers.

In 2021, Califrais became the official digital marketplace of Rungis with the launch of the platform *https://rungismarket.com*. This platform provides detailed information about products and services, allowing customers to easily place orders according to their needs. Figure 1.1 is taken from this website.

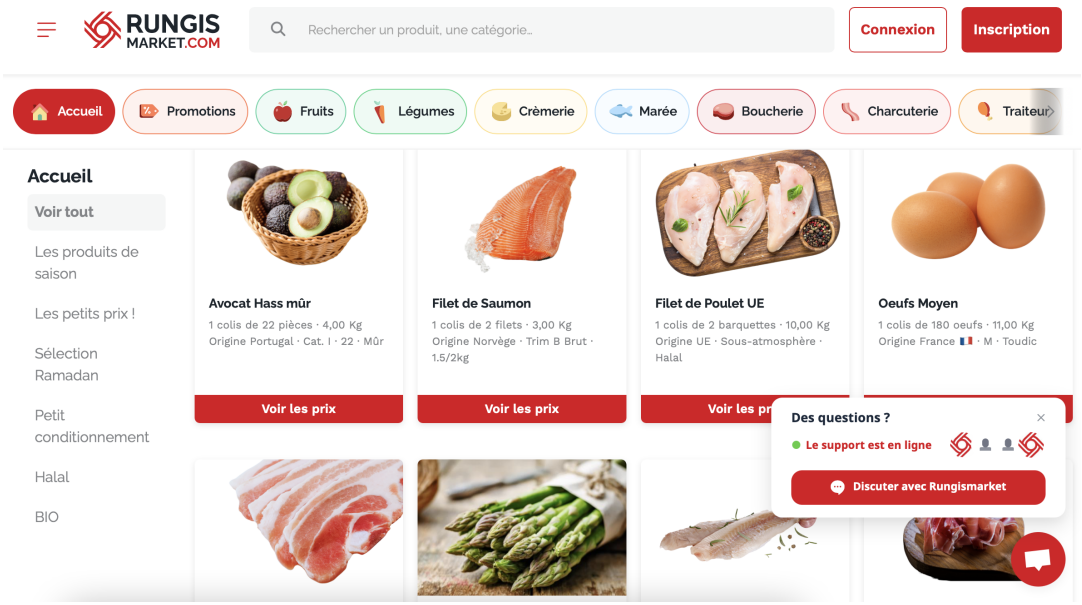

Figure 1.1 – A screenshot taken from Rungis website.

The entire supply chain is then managed internally to ensure that orders are delivered accurately and quickly. To achieve this efficiency, the company has developed customized tools to manage the flow of goods, warehouse inventory, product picking and sorting, order preparation, and delivery route optimization. Figure 1.2 shows a typical working day at Califrais.

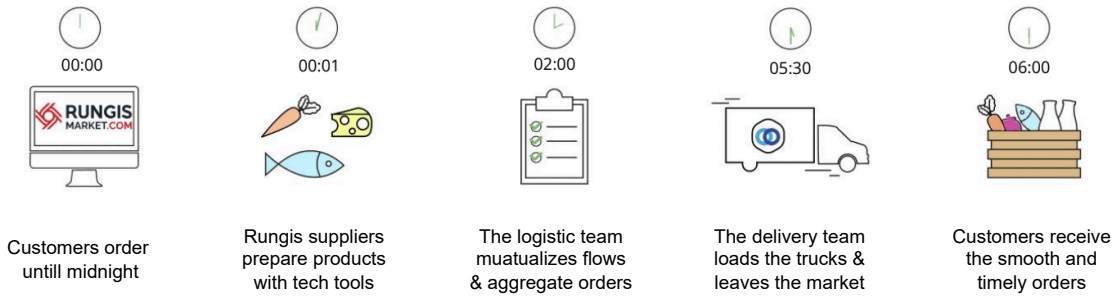

Figure 1.2 – Califrais operational process.

After a decade of creation and development, this start-up has successfully built a distri-

bution system with a catalog of more than 8000 products over 120 categories, expanding its customer base beyond Paris and internationally. To effectively maintain and develop this distribution system, the company has a system of specialized departments whose organizational structure is shown in Figure 1.3. Most of the research for this thesis was carried out within the Research team (LabCom department), with significant support from the Data team.

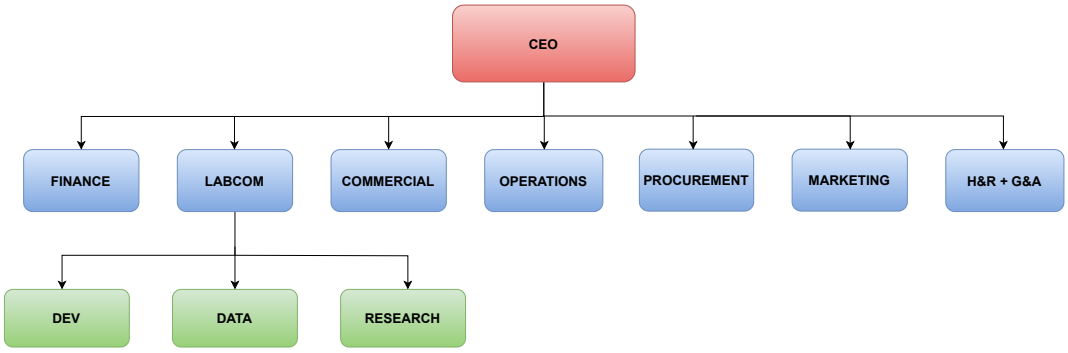

Figure 1.3 – Califrais organization structure

LabCom LOPF and the Machine Learning Research Team

To address current supply chain problems, as well as the rapid increase in product flows and customer acquisition at Califrais, the LabCom LOPF (Large-scale Optimization of Product Flows) was established in March 2021 as a public-private collaboration between Califrais and the Laboratoire de Probabilités, Statistique et Modélisation (LPSM), Université Paris Cité and Sorbonne Université. Currently, the LabCom LOPF has 13 members divided into three teams: the Dev team, the Data team, and the Research team, as illustrated in Figure 1.3. Each team has separate but complementary roles and functions. The Dev team manages and updates database systems, software, and applications. The Data team implements algorithms to solve the different data problems that Califrais faces. The Research team proposes new learning methods adapted to Califrais' specific problems. In the development of technological solutions, the lab has also received support thanks to strong partnerships with well-known research centers in France, including LPSM, Inria, and CNRS. A portion of the research for this thesis was conducted with the HeKa team at Inria Paris.

With access to a diverse, rich, and complex database from operational services, as well as the support of experienced researchers, the Research team's motivation is to enhance the value of this data by developing new machine learning algorithms for optimizing logistics flows on a very large scale. Flow prediction is a fundamental project within the laboratory, with several PhD students working on inventory optimization and others focusing on route optimization. The inventory optimization project aims to determine each day which products to order for the warehouse to maximize profit, minimize food waste, and avoid shortages. Similarly, route optimization seeks to decide each day how many vehicles the company needs and their route, with the goals of minimizing the number of vehicles (and CO2 emissions), maximizing their load rate, avoiding delivery delays, and improving

customer satisfaction. The overall goal of these projects is to coordinate small factors to maximize vehicle loads, minimize CO2 emissions, and reduce food waste.

Customer Satisfaction Prediction

Califrais' data has also enabled them to develop tools that model customer satisfaction and predict customer churn before it occurs.

For most companies, acquiring a new customer costs more than retaining an existing one. According to the study conducted by Mozer et al. (2000), marketing campaigns for retaining existing customers provide a better return on investment than putting efforts into attracting new customers. In addition, the success of Califrais' business model depends on regular significant orders, making it crucial to prevent customers from leaving the service. Therefore, churn prediction, which identifies customers likely to leave the service and potential reasons for their dissatisfaction, enables marketing teams to take appropriate action for each individual customer to maximize the chances of retention and ultimately increase the value of the company.

Customer churn studies have been conducted across various service sectors, for example, games (Periáñez et al., 2016), telecommunication (Gui, 2017), or finance (Larivière and Van den Poel, 2004). These studies on the churn analysis attempted to identify or predict in advance the risk that customers will churn and the reasons for their churns using various indicators. One of the most typical customer churn analysis indicators is the customer churn rate which refers to the ratio of customers who leave a service to the total number of customers during a specific period. In Califrais at the moment, monthly churn rates have been computed using a fixed one-month time window: a customer that does not place any new order beyond that limit is considered churned. Although this method has the crucial advantage of simplicity, it cannot help to identify customers at high risk of churn, nor can it explain the reasons for their churn.

The company is then looking for more effective solutions by focusing on building machine learning churn models based on a variety of features. A majority of them are extracted from several kinds of historical order data collected over multiple points in time from the same customer - also known as longitudinal data. These kinds of data can be listed as follows:

- The dates on which the orders were placed. It can help to extract information like the customer's acquisition date, their level of loyalty (e.g. number of orders placed per week, etc.), and the time between two orders.
- The content of the orders placed by a customer such as the products requested, the quantities, and the categories.
- The level of customer satisfaction derived from customer's comments and ratings which may be obtained after each order they place.
- The quality of the delivery such as the number of missing items and the lateness of each delivery.

For example, among the 3 restaurants A, B, and C shown in Figure 1.4, there are differences in the ordering pattern according to the number of products ordered and the number

of categories in which these products were ordered. While restaurant B churns after three months of service, restaurants A and C keep their service for a longer time.

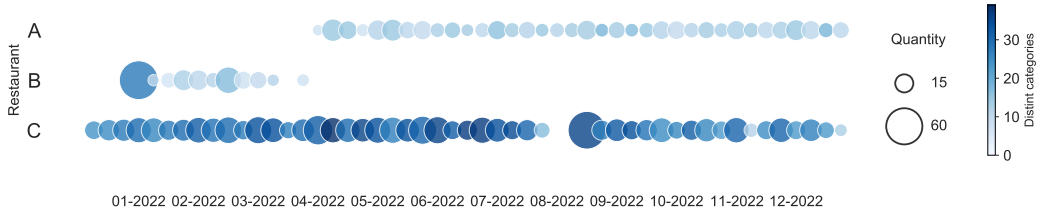

Figure 1.4 – Order history of 3 customers (A, B, and C), acquired in 2022, differing in the number of products ordered, and by the number of categories in which these products were ordered.

From classification to survival analysis for churn prediction

Most previous works transform the churn prediction into a binary classification, which predicts whether a customer will churn within a predefined time frame, see e.g. Buckinx and Van den Poel (2005), Coussement and Van den Poel (2008), Verbeke et al. (2012), and Zhang et al. (2017). This approach consists of labeling each client as a churner or not, allowing the problem to fall into the large domain of supervised learning, with many algorithms available. However, the labeling process can vary depending on the specific business context.

In contractual settings, churn typically refers to clients who do not renew their contracts when they expire, leading to predictions about whether a client will churn at the renewal date based on their historical activity, see e.g. Coussement and Van den Poel (2008) and Zhang et al. (2017).

In non-contractual settings, where no formal contract binds the client to the company, clients can leave at any time without restrictions. This is the case for Califrais. In such cases, a specific churn criterion is constructed. For instance, if a client stops using a service for a certain period, known as the churn window, they are considered a churn client. Figure 1.5 below shows an example of churn definition in the non-contractual case. The churn prediction then turns to the classic binary classification, similar to the contractual case mentioned above (see e.g. Buckinx and Van den Poel (2005) and Verbeke et al. (2012)).

Although this method simplifies the churn prediction problem, its performance highly depends on choosing the churn criterion in the non-contractual context. In addition, in many services, estimating the survival time, i.e. the time elapsed before the client churns, is critical for timely interventions and efficient resource allocation. However, this method cannot distinguish between those who churned at the beginning of the churn window and those who churned later, nor can it predict the survival time of customers who have not churned yet (Khodadadi et al., 2020).

To avoid the drawbacks of the classification approach, some methods have shifted towards estimating the time until a client churns. When the churning times of all customers

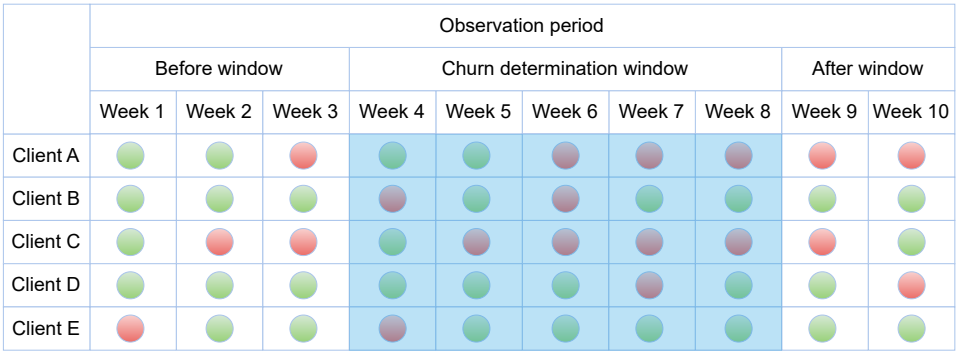

Figure 1.5 – Schematic of the time window method used for churn prediction in the binary approach. The data represents 5 clients over a 10-week period, where a red dot indicates no order was placed in a given week and a green dot signifies an order was placed. According to the churn definition, which considers a client to have churned if no orders are placed for 4 consecutive weeks, clients A and C are identified as having churned at weeks 6 and 5, respectively, while clients B, D, and E are censored at week 10. To illustrate the process of defining the churn label in the binary approach, the time window of length $\delta_t = 5$, which is in the blue area, is set from week 4 $(t = 4)$ to week 8 $(t + \delta_t = 8)$. Since clients A and C churn within this window, they are labeled as churned, while clients B, D, and E are labeled as non-churned.

are observed, standard regression models for continuous responses can be used to predict this time (Buis, 2006). However, in many cases - particularly at Califrais - churn observations are incomplete due to the limited duration of historical data or because customers are still actively engaging with the service, a situation referred to as censoring. Therefore, it is crucial to design a model that not only estimates the churning time but also addresses the censoring problem.

Survival analysis, a branch of statistics that studies the data representing the time until a specific event occurs - also known as time-to-event data or survival data, has recently emerged as a more flexible alternative for addressing the churn prediction problem in the presence of censored data (Larivière and Van den Poel, 2004; Periáñez et al., 2016; Bertens et al., 2017). This method was originally developed in the medical field to study patient survival times, but its applications have since expanded to other domains. For instance, in industry, a manufacturer may use survival analysis to estimate the lifespan of machine components, while in finance, a bank may use it to predict how long it will take a borrower to fully repay a loan. In survival analysis, the time until the event of interest - known as survival time - is considered a random variable whose distribution is generally described and modeled in terms of two related functions, the survivor function and the hazard function. The details of this framework are provided in the next chapter.

In the context of the churn prediction problem, the number of applied studies using this method has recently increased but remains limited (see Table 2 from the survey conducted by Ahn et al. (2020)). Most of these studies focus on games (Sifa et al., 2014; Periáñez et al., 2016; Bertens et al., 2017), and a few studies on other sectors such as finance and telecommunications (Larivière and Van den Poel, 2004), where different survival analysis frame-

works, such as Cox proportional hazards and random survival forests, have been applied. There are several reasons for this limitation: challenges in scaling with high-dimensional data, restrictive assumptions about the relationships between the variables and the event, or difficulties incorporating longitudinal data.

1.2 Extension of survival analysis with longitudinal data

Considering the limited application of survival models in churn prediction, especially due to the lack of longitudinal data integration, the following section highlights the importance of including such data and explores the challenges associated with this integration. Building a prediction model using only the prognostic features at baseline - also known as baseline features - may be suboptimal because it does not fully utilize the large amount of longitudinal data collected during the study. More recent measurements, which are temporally closer to the event of interest, may have a stronger association with the risk of the event. Therefore, incorporating these features into survival models leads to more accurate predictions of survival probabilities. This is also important for Califrais, where much of the historical customer order data is in the form of longitudinal data, making it critical to quantify the effect of this data on the survival time of customers.

In other sectors, such as healthcare, it is becoming increasingly common to record the values of longitudinal features (e.g. biomarkers such as heart rate or hemoglobin level) up to the occurrence of an event of interest, such as rehospitalization, relapse, or disease progression. These longitudinal features can also be crucial in predicting the time until an event occurs. For example, in the context of sepsis prediction, consider three patients, A, B, and C, who were admitted to the hospital for sepsis diagnosis. There are differences in their characteristics of heart rate and respiration rate. While the values of patient A are stable, the measurement of heart rate and respiration rate of patients B and C change a lot near the time-to-event.

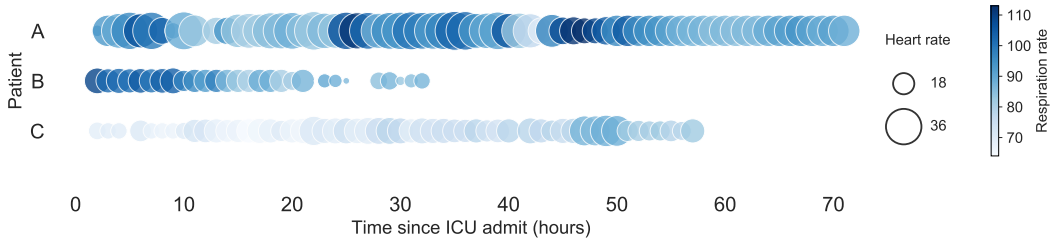

Figure 1.6 – Historical profiles of 3 patients (A, B, and C), differing in the heart rate and the respiration rate.

Despite its many advantages, incorporating longitudinal data into a survival model is not straightforward. There are several reasons for this. The longitudinal data, also known as repeated data, can have complex structures, such as repeated measures within individuals, leading to correlated observations. In addition, the data may follow non-linear trajectories that change over time, which may not be adequately captured by traditional modeling

techniques. Furthermore, joint modeling of longitudinal and survival data adds further complexities, such as model convergence issues, computational burden, and interpretation challenges. These challenges must be carefully addressed when developing a new framework for survival analysis with longitudinal data.

The survival analysis framework is closely related to the counting process. Although both approaches are used to analyze time-to-event data, the counting process focuses on counting the occurrences of events over time rather than modeling the time until the event occurs. The counting process involves modeling the intensity or rate at which events occur over time, taking into account the event times and the corresponding covariate information, either in static or longitudinal data. Survival analysis can be seen as a special case of the counting process, where the focus is on modeling the hazard function, which represents the instantaneous rate of occurrence of events given survival up to a certain time. Many statistical methods and models developed in one framework can be adapted or extended to the other, allowing for a unified approach to analyzing time-to-event data. Developing a new framework for the counting process is also interesting for Califrais where it can be applied to the problem of predicting the time when clients place their next order.

1.3 Objective of the thesis

This thesis aims to contribute to the development of new frameworks of survival analysis to predict the risk of an event in the presence of censored survival data and multivariate longitudinal data and its extension to counting processes; and the application of these frameworks to the churn prediction problem at Califrais.

Chapter 2

Survival analysis with longitudinal data

Contents

In this chapter, we describe the material of survival analysis and its relation to longitudinal data. First, in Section 2.1, we introduce concepts, tools, the general formalism of survival analysis, counting processes, and several survival learning models with baseline features. Section 2.2 reviews the previous works of survival analysis with longitudinal data. In Section 2.3, we expand our discussion of the more advanced framework, focusing on deep survival methods with longitudinal data. Section 2.4 introduces evaluation metrics

for assessing the prediction performance in survival analysis. Finally, Section 2.5 outlines the contributions of this thesis.

2.1 Survival analysis

2.1.1 Survival Function and Hazard rate

Let $\tilde{T}$ be a non-negative random variable modeling the time until the event of interest occurs, which is called the survival time of an individual. We denote its cumulative distribution function and probability density function by F and f respectively. In survival analysis, we are typically more interested in predicting how long an individual will survive, rather than how quickly its event of interest will occur, which is described by the survival function. We then describe the survival function and the related hazard function in the following two definitions, as they are fundamental to the distribution of $\tilde{T}$ and are essential for making predictions in survival analysis.

Definition 2.1.1. The survival function, denoted by S, corresponds to the probability that the event of interest did not occur at time $t \geq 0$, and is given by

$$S(t) = \mathbb{P}(\tilde{T} > t) = 1 - F(t). \tag{2.1}$$

Definition 2.1.2. The hazard function, denoted by λ, corresponds to the infinitesimal probability of the event of interest occurring just after time t, conditional on having survived at least until time t, and is expressed as follows

$$\lambda(t) = \lim_{dt \to 0} \frac{\mathbb{P}(t \leq \tilde{T} < t + dt | \tilde{T} \geq t)}{dt} = \lim_{dt \to 0} \frac{\mathbb{P}(t \leq \tilde{T} < t + dt)}{S(t)dt} = \frac{f(t)}{S(t)}. \tag{2.2}$$

Combining (2.1) and (2.2), we obtain

$$\lambda(t) = \frac{dF(t)}{S(t)} = \frac{-dS(t)}{S(t)} = -d \log S(t),$$

so that, given that $S(0) = 1$, the survival function S can be rewritten as

$$S(t) = \exp\Big(- \int_0^t \lambda(u) du \Big).$$

2.1.2 Censoring and Likelihood

Censoring. As mentioned in the previous chapter, the specific difficulty in survival analysis is the possible existence of censoring, where the survival times of some individuals

are unknown because they have not experienced the event during the study. This phenomenon occurs in different ways. For example, individuals may leave the study at any time without experiencing the event, or they may not have experienced the event by the end of the study (right-censored). The event may also have occurred before the start of the study (left-censored). In this thesis, we only consider right-censored individuals. In this case, what we know about the survival time $\tilde{T}$ is that it exceeds the time counted until the individuals leave the study or the end of the study. This time is called the time to censoring and is denoted by C.

In the presence of censoring in the study, an individual can either experience the event or be censored. The time we actually observe, called observed time and denoted by T, can then be the survival time $\tilde{T}$ or the time to censoring C. We observe $\tilde{T}$ only when $\tilde{T} \leq C$, otherwise we only know that $\tilde{T} > C$. The observed time T is then defined as

$$T = \tilde{T} \wedge C,$$

where $a \wedge b$ denotes the minimum between two real numbers a and b.

A corresponding censoring indicator which is required to distinguish whether the observed time is the survival time or the time to censoring, is denoted by Δ and defined as

$$\Delta = \mathbb{1}_{\{\tilde{T} \leq C\}}.$$

For each individual, we observe a pair of observed time and censoring indicator (T, Δ). The observed dataset for n individuals, which are assumed to be independent and identically distributed (i.i.d.), is then represented as

$$\mathcal{D}_n = \{(T_1, \Delta_1), (T_2, \Delta_2), \ldots, (T_n, \Delta_n))\}.$$

Likelihood. The estimate of the desired quantity (i.e. survival function, hazard function) can be obtained by maximizing the likelihood function defined on the dataset $\mathcal{D}_n$. We first write the likelihood function L_i for an individual $i \in \{1, \ldots, n\}$. If this individual is non-censored ($\Delta_i = 1$), we know that the survival time is exactly T_i. The contribution to the likelihood of this individual is therefore the value of the density function at T_i, that is,

$$L_i = f(T_i).$$

Otherwise, if this individual is censored ($\Delta_i = 0$), we know that the survival time is at least T_i. The contribution to the likelihood of this individual is then the value of the survival function at T_i, that is,

$$L_i = S(T_i).$$

By removing the constants, the part of the likelihood L involving functions character-

izing the distribution of $\tilde{T}$ of the entire sample $\mathcal{D}_n$ can be written

$$L = \prod_{i=1}^{n} L_i = \prod_{i=1}^{n} f(T_i)^{\Delta_i} S(T_i)^{1-\Delta_i} = \prod_{i=1}^{n} \Big(\frac{f(T_i)}{S(T_i)}\Big)^{\Delta_i} S(T_i) = \prod_{i=1}^{n} \lambda(T_i)^{\Delta_i} S(T_i),$$

and the log-likelihood $\mathcal{L}$ as

$$\mathcal{L} = \log L = \sum_{i=1}^{n} \Delta_i \log \lambda(T_i) + \log S(T_i) = \sum_{i=1}^{n} \Delta_i \log \lambda(T_i) - \int_0^{T_i} \lambda(u) du. \tag{2.3}$$

2.1.3 Counting Processes

Since the seminal work of Aalen in the 1970's (Aalen, 1978), it has been recognized that survival analysis can be cast in the general theory of counting processes. For a historical review, we refer the reader to Aalen et al. (2010) and we briefly outline here the main ideas of this theory. Suppose that an individual can experience several events $0 \leq T_1 < T_2 < \dots$. The observation of these event times is equivalent to the observation of the counting process

$$\tilde{N}(t) = \sum_{j \geq 1} \mathbb{1}_{T_j \leq t}.$$

Note that this stochastic process has piecewise constant trajectories with jumps of size $+1$ at each time T_k and generates the natural historical filtration $\mathcal{F} = (\mathcal{F}_t)_{t \geq 0}$ where $\mathcal{F}_t = \sigma(\tilde{N}(s),\, s \leq t)$ is a σ-algebra. Central to statistical inference in this theory is that a counting process, as a submartingale, admits a Doob-Meier decomposition. This means that there exists an unique predictable process Λ, called the compensator of $\tilde{N}$, such that the process $\tilde{N} - \Lambda$ is a martingale, that is,

$$\mathbb{E}(\tilde{N}(t) - \Lambda(t) | \mathcal{F}_s) = \tilde{N}(s) - \Lambda(s)$$

for all $s \leq t$. It is this connection with martingale theory that has made the work of Aalen and colleagues so fruitful, see Andersen et al. (2012).

The derivative of the compensator Λ (when it exists) is called the intensity and measures the infinitesimal probability of an event occurring at time t knowing the history up to that time

$$\lambda(t) = \lim_{h \to 0} \frac{1}{h} \mathbb{E}(\tilde{N}(t+h) - \tilde{N}(t) | \mathcal{F}_t) = \lim_{h \to 0} \frac{1}{h} \mathbb{P}(\tilde{N}(t+h) - \tilde{N}(t) = 1 | \mathcal{F}_t)).$$

Note that this intensity function is an adapted version of the one defined in (2.1) for the counting process. As the functions Λ and λ determine the distribution of the counting process $\tilde{N}$, statistical inference focuses on estimating one of them. In addition, we can integrate the phenomenon of right censoring at time C into this model by introducing a

process Y, known as the at-risk process and defined as

$$Y(t) = \mathbb{1}_{C \geq t}.$$

When Y is null, the jumps in $\tilde{N}$ are not observed, the observed counting is then

$$N(t) = \tilde{N}(t \wedge C) = \int_0^t Y(s) d\tilde{N}(s) = \sum_{k \geq 1} Y(T_k) \mathbb{1}_{T_k \leq t}.$$

It can be shown (see Andersen et al., 2012) that the intensity of N is given by λY and that the log-likelihood associated with the observations $\{N(s), Y(s), 0 \leq s \leq \tau\}$ over a period of study ending at time τ, is given by

$$\ell(\lambda) = \int_0^\tau \log \lambda(s) dN(s) - \int_0^\tau \lambda(s) Y(s) ds.$$

If one considers that one cannot observe events past the first, the at-risk process takes the form

$$Y(t) = \mathbb{1}_{C \wedge \tilde{T} \geq t}.$$

This corresponds to the classic survival analysis settings presented above. In this case, the intensity is equal to the hazard function.

2.1.4 Discrete survival time

In the literature, most methods deal with continuous survival times, as discussed in the previous section. However, some frameworks consider survival time to be discrete. This approach arises from the fact that time is often observed in discrete units (days, months, years, etc.), or from continuous time grouped into contiguous intervals and treated as discrete. We now present several functions to describe the distribution of a discrete survival time. In general, the survival time is assumed to be in a set of J positive incremental discrete times $\{\tau_1, \ldots, \tau_J\}$. The probability mass function and the survival function for the survival time $\tilde{T}$ are defined respectively as

$$f(\tau_j) = \mathbb{P}(\tilde{T} = \tau_j), \tag{2.4}$$

and

$$S(\tau_j) = \mathbb{P}(\tilde{T} > \tau_j) = \sum_{j' \,:\, \tau_{j'} > \tau_j} f(\tau_{j'}). \tag{2.5}$$

The hazard function is still a key quantity to estimate in this approach and is defined as

$$\lambda(\tau_j) = \mathbb{P}(\tilde{T} = \tau_j \,|\, \tilde{T} \geq \tau_j) = \frac{f(\tau_j)}{S(\tau_{j-1})} = \frac{S(\tau_{j-1}) - S(\tau_j)}{S(\tau_{j-1})}. \tag{2.6}$$

A consequence of the previous equation is that the survival function can be rewritten as

$$S(\tau_j) = \big(1 - \lambda(\tau_j)\big) S(\tau_{j-1}) = \prod_{j'=1}^{j} \big(1 - \lambda(\tau_{j'})\big).$$

The log-likelihood function $\mathcal{L}$ for discrete survival time can be written in two forms (Tutz, Schmid, et al., 2016), which are

$$\mathcal{L} = \sum_{i=1}^{n} \Delta_i \log\big(f(T_i)\big) + (1 - \Delta_i) \log\big(S(T_i)\big) \tag{2.7}$$

$$= \sum_{i=1}^{n} \Delta_i \log\big(\lambda(T_i)\big) + (1 - \Delta_i) \log\big(1 - \lambda(T_i)\big) + \sum_{j:\tau_j < T_i} \log\big(1 - \lambda(\tau_j)\big). \tag{2.8}$$

Aside from the fact that survival data are often recorded at discrete intervals, making the discrete survival model more suitable and representative of the data collection process, this approach also provides several benefits, such as flexible modeling and the straightforward interpretation of the hazard function, which can be formulated as conditional probabilities (Tutz, Schmid, et al., 2016). However, the decision between using discrete or continuous time survival models can depend on various factors, including the nature of the data, the computational feasibility, and whether the focus is on modeling survival time intervals or exact survival times.

2.1.5 Learning with baseline features

Having fully introduced the core concepts of survival analysis in the previous sections, we now explore a variety of survival frameworks, ranging from classical linear statistical methods to more advanced machine learning and recent deep learning approaches. These frameworks accommodate different types of features, including the baseline features whose values are fixed over time, as well as longitudinal data.

In this section, we briefly describe different methods for modeling the effect of baseline features on the distribution of the survival time. The other frameworks that deal with the longitudinal data are discussed in the next section. We first introduce the Cox proportional hazard model (Cox, 1972b), which is a classic and widely used method in survival analysis. We then present the random survival forests model (Ishwaran et al., 2008), which is an extension of the classical random forests (Breiman, 2001) to handle time-to-event data. We end this section by describing several deep survival frameworks, which are the most recent methods and are extensions of modern deep learning architectures to handle the time-to-event data.

For each individual $i \in \{1, \ldots, n\}$, besides the observed time T_i and the censoring indicator Δ_i introduced previously, we denote by $W_i \in \mathbb{R}^p$ the $p-$dimensional vector of

baseline features. The dataset is updated as

$$\mathcal{D}_n = \{(T_1, \Delta_1, W_1), \ldots, (T_n, \Delta_n, W_n)\}.$$

Cox proportional hazards model

A proportional hazards model assumes that the hazard rate (or intensity) of an individual at any time is proportional to a baseline hazard function, which can vary over time, multiplied by a function of features - also known as the risk function. Mathematically, for an individual i with baseline features W_i, the hazard function at time t is given by

$$\lambda_i(t \,|\, W_i) = \lambda_0(t) \exp\big(h(W_i)\big), \tag{2.9}$$

where $\lambda_0(t)$ is an unknown baseline hazard function common to all individuals and $h(W_i)$ is the risk function (or marker function not to be confused with the loss function) denoting the effects of individual covariates W_i.

The Cox proportional hazard (Cox-PH) model (Cox, 1972b) is the proportional hazards model that represents the risk function $h_\alpha(W_i)$ for an individual i by a linear function of the baseline features W_i and their associated coefficients α, which is $h_\alpha(W_i) = W_i^\top \alpha$. Cox (1972b) proposed to estimate the coefficients $\alpha \in \mathbb{R}^p$ by maximizing the partial log-likelihood which is defined as

$$\ell(\alpha) = \sum_{i=1}^{n} \Delta_i \big(W_i^T \alpha - \log \sum_{j:T_j \geq T_i} \exp(W_j^T \alpha) \big). \tag{2.10}$$

Let us denote by $\hat{\alpha} = \text{argmax } \ell(\alpha)$ the maximum likelihood estimator. In addition, Breslow (1972) suggested to estimate the cumulative baseline hazard function $\Lambda_0(t) = \int_0^t \lambda_0(u)du$ by

$$\hat{\Lambda}_0(t) = \sum_{i=1}^{n} \frac{\mathbb{1}_{T_i \leq t} \Delta_i}{\sum_{j:T_j \geq T_i} \exp(W_j^T \hat{\alpha})}, \tag{2.11}$$

which is now known as the Breslow estimator, see Andersen et al. (2012). It turns out that $\hat{\alpha}$ and $\hat{\Lambda}_0(t) = \int_0^t \hat{\lambda}_0(u)du$ can be seen as the maximum likelihood estimators of the complete likelihood given by

$$\mathcal{L}(\alpha, \lambda_0(.)) = \sum_{i=1}^{n} \Delta_i \big(\log \lambda_0(T_i) + W_i^T \alpha \big) - \int_0^{T_i} \lambda_0(u) \exp(W_j^T \alpha) du. \tag{2.12}$$

The corresponding estimated survival function for an individual i is

$$\hat{S}_i(t) = \exp\big(- \hat{\Lambda}_0(t) \exp(W_i^\top \hat{\alpha}) \big). \tag{2.13}$$

Although this method is simple and easy to interpret, it has significant limitations. Its assumptions can be easily violated in cases where the effects of covariates are non-linear or

change over time. In addition, if there are interactions between covariates, the assumption that their effects are additive does not hold (Van Houwelingen, 2007).

Several alternative methods have been proposed to address the limitations of the Cox model. Below we describe two families of methods that aim to overcome the assumption of linearity in the risk function.

Survival Random Forest

Proposed by Ishwaran et al. (2008), random survival forest is an ensemble learning method for survival analysis that extends the concept of random forest (Breiman, 2001) to handle time-to-event data. The random survival forest generally follows the same principles as the random forest. However, this framework differs from the random forest in two main aspects: it uses the log-rank statistic (Bland and Altman, 2004) as the splitting rule, and it estimates the survival or cumulative hazard function at the terminal node. These two aspects are described in detail below.

Log-rank splitting rules. This splitting rule aims to maximize the survival difference between the two resulting nodes after the split. Given a set H of individuals at a specific tree node to be split, we suppose that a proposed feature $W^q \in \{W^1, \ldots, W^p\}$ (one of the coordinates of W) with a splitting value c is used to split the node into left and right child, $H^l = \{i : i \in S, W_i^q \leq c\}$ and $H^r = \{i : i \in S, W_i^q > c\}$ respectively. Let $\{\tau_1, \ldots, \tau_J\}$ be the set of distinct failure times taken from H. At the time $\tau_j \in \{\tau_1, \ldots, \tau_J\}$, we denote by Y_j^l and Y_j^r the number of individuals still at risk in the left and right child nodes respectively, which are given by

$$Y_j^l = \sum_{i \in H^l} \mathbb{1}_{T_i \geq \tau_j} \quad \text{and} \quad Y_j^r = \sum_{i \in H^r} \mathbb{1}_{T_i \geq \tau_j}.$$

Similarly, d_j^l and d_j^r are the numbers of individuals for whom the event has occurred at time τ_j in the left and right child node respectively, that are,

$$d_j^l = \sum_{i \in S^l} \mathbb{1}_{T_i = \tau_j} \quad \text{and} \quad d_j^r = \sum_{i \in S^r} \mathbb{1}_{T_i = \tau_j}.$$

Finally, denoting by $Y_j = Y_j^l + Y_j^r$ the total number of individuals still at risk and $d_j = d_j^l + d_j^r$ the total number of individuals for whom the event has occurred, the log-rank split-statistic value for the split is defined as

$$L(W^q, c) = \frac{\sum_{j=1}^{J} \left(d_j^l - d_j \frac{Y_j^l}{Y_j}\right)}{\sqrt{\sum_{j=1}^{J} \frac{Y_j^l}{Y_j}\left(1 - \frac{Y_j^l}{Y_j}\right)\left(\frac{Y_j - d_j}{Y_j - 1}\right) d_j}}.$$

This statistic value quantifies the difference between the observed and expected number of events across the two nodes after the split, with a larger value indicating a greater difference

in survival curves between the two nodes. The split is then performed for the values q and c that maximize the statistic value.

Estimation at terminal node. For each tree, the survival function S or the cumulative hazard function Λ are estimated at each terminal node. Let h be a terminal node in the survival tree and $\{\tau_1^h, \ldots, \tau_{K_h}^h\}$ be the set of distinct failure times taken from individuals in that node. For each time $\tau_j^h \in \{\tau_1^h, \ldots, \tau_{K_h}^h\}$ at node h, we denote by Y_j^h the number of individuals still at risk and by d_j^h the number of individuals for whom the event has occurred at that time. For a survival tree $m \in \{1, \ldots, M\}$ in a random survival forest model of M trees, the estimation of the survival function and cumulative hazard function at the terminal node h are given respectively by

$$S_m^h(t) = \prod_{\tau_j^h \leq t} (1 - \frac{d_j^h}{Y_j^h}) \quad \text{and} \quad \Lambda_m^h(t) = \sum_{\tau_j^h \leq t} \frac{d_j^h}{Y_j^h},$$

which are the Kaplan-Meier (Kaplan and Meier, 1958) and Nelson-Aalen (Aalen, 1978) estimators. The estimations from the associated terminal nodes of all trees are then averaged to obtain the final estimation.

The advantages of the random survival forest are its high flexibility, its non-linear and non-parametric structure, and easy scalability to high-dimensional problems. However, its disadvantage is similar to that of the random forest, as it tends to favor variables with numerous split values, leading to bias in the resulting summary estimations (Nasejje et al., 2017).

Deep survival model

Another line of work that aims to incorporate non-linearities involves deep learning. Below we describe deep survival models, which combine deep neural network techniques (LeCun et al., 2015; Goodfellow et al., 2016) with survival analysis to model time-to-event data. This kind of approach offers several advantages such as the ability to automatically learn complex patterns and interactions from high-dimensional data, and the flexibility to handle non-linear or time-varying relationships in the survival model (Katzman et al., 2018).

The architecture of deep neural networks can be composed of various types of layers, such as fully connected layers, convolution layers, recurrent layers, or softmax layers, depending on the type of input data (i.e. tabular data, image, text, or longitudinal data), the type of estimated output (i.e. risk function, hazard function or density function), or the specific task (i.e. single risk, competing risk, or complex multistate models), see Kalbfleisch and Prentice (2002) for a presentation.

Figure 2.1 shows an example of a fully connected network architecture with L hidden layers that handle the baseline features W in the input layer and derive the function ϕ at the output layer whose shape (single node or multiple nodes) depends on the function to be estimated (risk function, hazard function or density function).

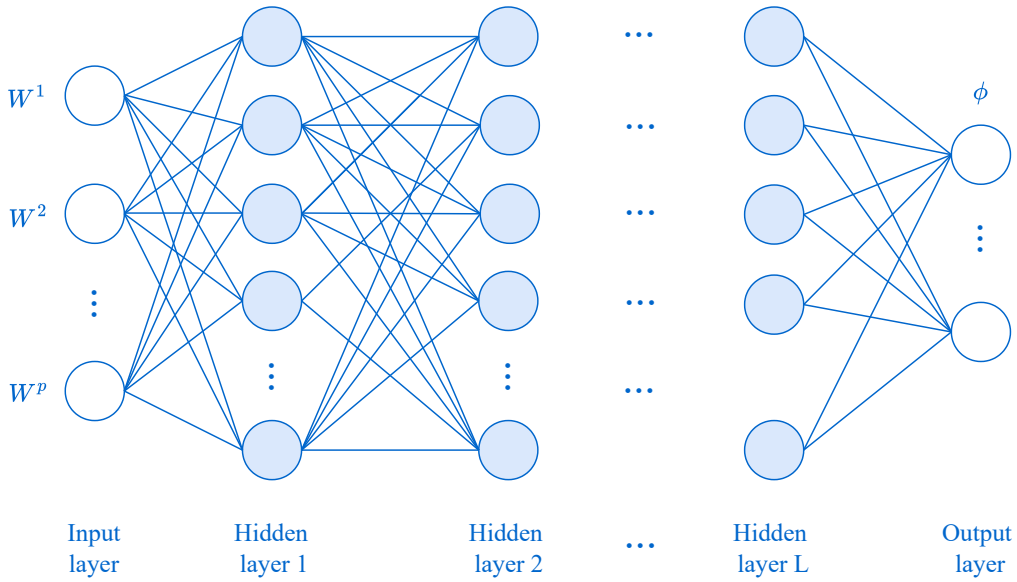

Figure 2.1 – An illustration of a fully connected deep network architecture that processes the baseline features W in the input layer, with the output layer capable of representing risk function, hazard function, or density function.

To incorporate baseline features in the model, there is a variety of deep survival frameworks that have been developed from classical feedforward neural network with only one hidden layer (Faraggi and Simon, 1995) up to most recent deep network techniques (Katzman et al., 2018; Tong and Zhao, 2022). We refer the reader to Wiegrebe et al. (2024) for a more detailed review.

These algorithms can be classified into two main approaches: the Cox-based approach and the discrete-time approach (Kvamme and Borgan, 2021). The Cox-based approach, which is considered an extension of the Cox regression model, represents the risk function through a neural network and optimizes the partial log-likelihood of the Cox-PH model (Faraggi and Simon, 1995; Yousefi et al., 2017; Katzman et al., 2018; Tong and Zhao, 2022). On the other hand, the discrete-time approach considers the survival time to be discrete and uses a neural network to model the discrete hazard function or probability mass function (Biganzoli et al., 1998; Fotso, 2018; Lee et al., 2018; Gensheimer and Narasimhan, 2019; Kvamme and Borgan, 2021).

Cox-based approach. Faraggi and Simon (1995) extended the Cox-PH model by parameterizing the risk function $h(W)$ in (2.9) with a neural network in the form of one hidden layer and a single node output. Figure 2.2 below shows the neural network architecture in the framework proposed by Faraggi and Simon (1995). We let θ be the weight of the neural network. For an individual i with the baseline features W_i, the output $\phi_\theta(W_i)$ represents the risk function $h(W_i)$, which is given given by

$$h(W_i) = \phi_\theta(W_i).$$

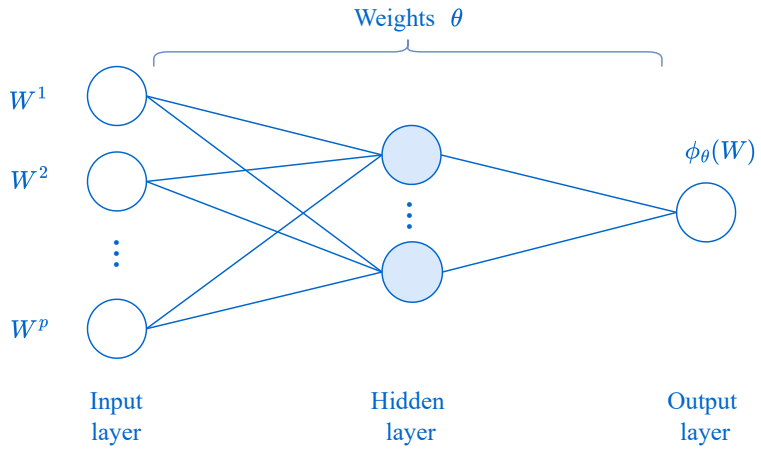

Figure 2.2 – The neural network architecture with one hidden layer and a single node output layer. This network, whose weight is denoted by θ, takes the baseline feature W as input and outputs $\phi_\theta(W)$ which is the estimation of the risk function $h(W)$.

The loss function $\mathcal{L}^{\text{Faraggi et al.}}(\theta)$ is set to be similar to the partial log-likelihood (2.10) with ℓ_2-regularization, which is

$$\mathcal{L}^{\text{Faraggi et al.}}(\theta) = \sum_{i=1}^{n} \Delta_i \Big[\phi_\theta(W_i) - \log \sum_{j:T_j \geq T_i} \exp\big(\phi_\theta(W_j)\big)\Big] + ||\theta||_2^2. \tag{2.14}$$

Yousefi et al. (2017) extends the network of Faraggi and Simon (1995) by using a deep neural network architecture and Bayesian optimization methods to optimize the network hyper-parameters (i.e. the number of layers, the number of nodes in each layer, ...). The framework *DeepSurv* proposed by Katzman et al. (2018) is another extension of the model of Faraggi and Simon (1995), incorporating modern architecture and optimization methods. In this framework, the network is in the form of a deep feedforward neural network with different non-linear hidden layer activation functions (Rectified Linear Units, Scaled Exponential Linear Unit). The main new feature in *DeepSurv* is the use of adapted stochastic gradient descent (Bottou, 2012) for optimizing the loss function (2.14). Indeed, due to the presence of the sum over the entire risk set

$$\sum_{j:T_j \geq T_i} \exp\big(\phi_\theta(W_j)\big)$$

in the partial likelihood of Equation (2.14), modern stochastic gradient descent optimization is not as effective as in other settings, such as regression or classification. To overcome this problem, *DeepSurv* uses a restricted risk set that only includes individuals in the current batch. Tong and Zhao (2022) propose *NN-DeepSurv*, which extends *DeepSurv* by employing nuclear norm (Candes and Recht, 2012) for imputing missing features.

Discrete-time approach. In another line of work, the authors proposed to build up on the variety of existing architectures for classification tasks. Towards that end, they use the discrete-time model, as presented in Section 2.1.4. Biganzoli et al. (1998) proposes *PLANN* (partial logistic artificial neural network), a modification of Faraggi and Simon (1995), whose output layer consists of J nodes - denoted by $\{\phi_{\theta^1}, \ldots, \phi_{\theta^J}\}$ - with sigmoid activation functions to estimate the discrete hazard function at J discrete times $\{\tau_1, \ldots, \tau_J\}$. Figure 2.3 below shows the neural network architecture of the *PLANN* framework.

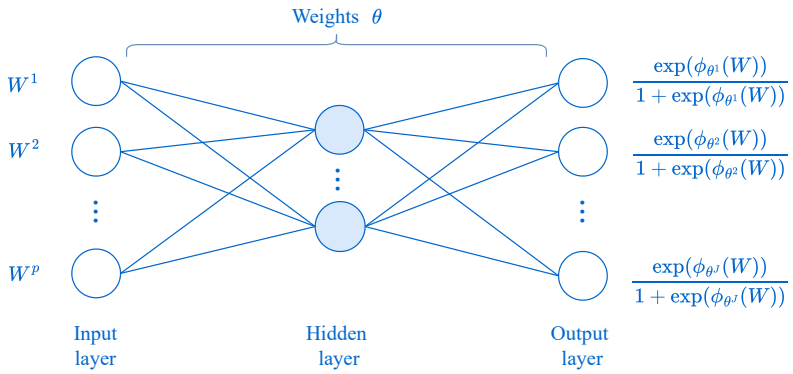

Figure 2.3 – The neural network architecture with one hidden layer and an output layer of J nodes. Each node $j \in \{1, \ldots, J\}$ is activated by the sigmoid activation function to estimate the discrete hazard function at time τ_j.

For an individual i with the baseline feature W_i, the output at each node $j \in \{1, \ldots, J\}$ is defined as

$$\lambda(\tau_j \,|\, W_i) = \mathbb{P}(\tilde{T}_i = \tau_j \,|\, \tilde{T}_i \geq \tau_j, W_i) = \frac{\exp(\phi_{\theta^j}(W_i))}{1 + \exp(\phi_{\theta^j}(W_i))}.$$

Parameters $\theta = (\theta^1, \ldots, \theta^J)$ are estimated by optimizing the likelihood function, defined similarly to (2.8) with the above estimation of the discrete hazard function. *Nnet-survival* (Gensheimer and Narasimhan, 2019) builds upon the work of Biganzoli et al. (1998), which is structured as a deep neural network with support for the convolution layer.

Fotso (2018) proposed *N-MTLR*, which is an extension of *MLLR* (muti-task logistic regression) (Yu et al., 2011). This framework uses a deep neural network whose output layer consists of J nodes - denoted by $\{\phi_{\theta^1}, \ldots, \phi_{\theta^J}\}$ - with a softmax activation function to parameterize the probability mass function at J discrete times $\{\tau_1, \ldots, \tau_J\}$. For an individual i with the baseline feature W_i, the output at each node $j \in \{1, \ldots, J\}$ is defined as

$$f(\tau_j \,|\, W_i) = \mathbb{P}(\tilde{T}_i = \tau_j \,|\, W_i) = \frac{\exp\left(\phi_{\theta^j}(W_i)\right)}{\sum_{j'=1}^{J} \exp\left(\phi_{\theta^{j'}}(W_i)\right)}.$$

The estimation of the discrete survival function S, defined similarly to (2.5), is based on the above estimation of the probability mass function. The parameters $\theta = (\theta^1, \ldots, \theta^J)$ are es-

timated by optimizing the likelihood function, defined similarly to (2.7) with the estimation of f and S.

Finally, *DeepHit* (Lee et al., 2018), which is a variant of *N-MTLR* (Fotso, 2018), consists of two subnetworks. The first subnetwork takes the baseline feature W as input and derives the output $\phi_{\theta_1}(W)$ to capture a latent representation, where θ_1 denotes the weight of this subnetwork. The second subnetwork, whose output layer is similar to *N-MTLR*, takes a pair $(\phi_{\theta_1}(W), W)$ as input and derives the probability mass function. For an individual i with the baseline feature W_i, the output at each node $j \in \{1, \ldots, J\}$ is defined as

$$f(\tau_j \,|\, W_i) = \mathbb{P}(\tilde{T}_i = \tau_j \,|\, W_i) = \frac{\exp\left(\phi_{\theta_2^j}(\phi_{\theta_1}(W_i), W_i)\right)}{\sum_{j'=1}^{J} \exp\left(\phi_{\theta_2^{j'}}(\phi_{\theta_1}(W_i), W_i)\right)},$$

where $\theta_2 = (\theta_2^1, \ldots, \theta_2^J)$ denotes the weight of the second subnetwork. The estimation of the discrete survival function S, defined similarly to (2.5), is based on the estimation of the probability mass function above. The loss function of *DeepHit* combines two types of loss which are the log-likelihood (similar to (2.7)) and a ranking loss for improving discriminative performance, which is as follows

$$\begin{aligned}\mathcal{L}^{\text{DeepHit}}(\theta) = &\sum_{i=1}^{n} \Big[(1 - \Delta_i) \log\left(S_\theta(T_i \,|\, W_i)\right) + \Delta_i \log\left(f_\theta(T_i \,|\, W_i)\right)\Big] \\ &+ \alpha \sum_{i=1}^{n} \sum_{\substack{j \neq i \\ j=1}}^{n} \mathbb{1}_{T_i < T_j} \Delta_i \eta\big(F_\theta(T_i \,|\, W_i), F_\theta(T_j \,|\, W_j)\big),\end{aligned}$$

where α are hyper-parameters chosen to trade off ranking losses, and $\eta()$ is a convex loss function. The architecture can handle competing risks when the event of interest may be due to two or more causes, see Kalbfleisch and Prentice (2002). We refer the reader to Lee et al. (2018) for a more detailed description.

2.2 Integrating longitudinal data

While the previous section concentrated on survival frameworks that address baseline features, we now shift our focus to other frameworks designed to handle longitudinal data. These include landmark methods, joint models, deep survival frameworks, and various feature extraction techniques for processing longitudinal data.

2.2.1 Introduction

We now consider that we obverse, for each individual $i \in \{1, \ldots, n\}$, in addition to the baseline features, a set of d longitudinal features at n_i time points $t_i^1 \leq \ldots \leq t_i^{n_i} \leq T_i$ up

to its observed time T_i. We let

$$X_i(t_i^j) = \big(X_{i1}(t_i^j), \ldots, X_{id}(t_i^j)\big) \in \mathbb{R}^d$$

be the vector of d observed longitudinal features at time $t_i^j \in \{t_i^1, \ldots, t_i^{n_i}\}$ and

$$X_i = \big(X_i(t_i^1), \ldots, X_i(t_i^{n_i})\big) \in \mathbb{R}^{d \times n_i}$$

be the entire history of the observed longitudinal marker up to its observed time T_i. Note that the terms "longitudinal data", "longitudinal features", and "longitudinal markers" are used interchangeably to refer to X_i. The observed longitudinal marker X_i is assumed to be the discretization, which can be prone to measurement errors, of an unobserved continuous process $x_i : [0, \tau] \to \mathbb{R}^d$, where τ is the time at the end of the study. In Figure 2.4 below, we show the longitudinal continuous processes (with $d = 1$) and their respective observed longitudinal markers for 4 individuals.

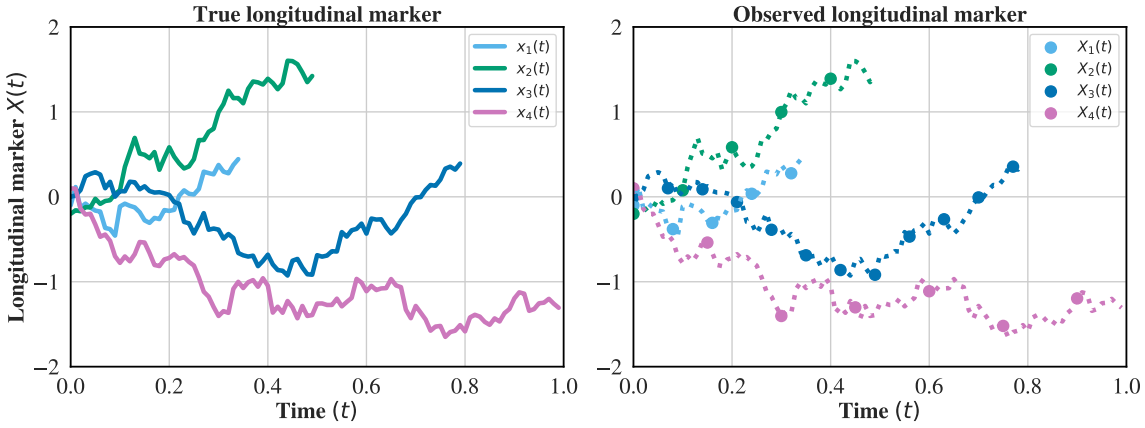

Figure 2.4 – The true longitudinal process (**left**) and the observed longitudinal data in • markers (**right**) for 4 individuals with $\tau = 1$. Each individual has its survival time equal to the duration up to its true last measurement time ($T_1 = 0.35, T_2 = 0.5, T_3 = 0.8, T_4 = 1.0$).

We now have the dataset

$$\mathcal{D}_n = \{(T_1, \Delta_1, W_1, X_1), \ldots, (T_n, \Delta_n, W_n, X_n)\}.$$

With this dataset, there are two main (and related) learning tasks that can be of interest: the estimation of the distribution of the survival time and the prediction of the remaining survival time at a given prediction time t_{P}. This prediction provides valuable information in many applications. For example, it identifies clients who are at higher risk of churning, helping the business to take appropriate actions or promotions, or it helps clinicians assess prognosis and determine personalized treatment plans.

Model estimation. Given a specified framework, we denote by θ the entire parameters that describe the relationship between the covariates - including both baseline and longi-

tudinal features - and the survival time distribution, which may take the form of a hazard function or density function. The optimal value of θ is determined by optimizing a chosen loss function, typically the likelihood or partial likelihood, using the training data. We then find the optimal value of θ by optimizing a loss function (typically, the likelihood or partial likelihood) on the training set.

Survival prediction. Given a trained model, we are interested in predicting the survival probability of a new individual i in the test set that survives up to the prediction time t_{P}. This prediction is based on the longitudinal measurements recorded up to time t_{P}, denoted by $X_{i,[0,t_{\mathrm{P}}]} = \left(X_i\left(t_i^1\right), \ldots, X_i\left(t_i^{n_i(t_{\mathrm{P}})}\right)\right)$, where $t_i^{n_i(t_{\mathrm{P}})}$ is the last recording time for longitudinal features before t_{P}. For any time $t > t_{\mathrm{P}}$, the probability that this new subject i will survive at least up to t, given that it has survived up to t_{P} with a set of longitudinal measurements $X_{i,[0,t_{\mathrm{P}}]}$, writes

$$\pi_i(t \,|\, t_{\mathrm{P}}) = \mathbb{P}(\tilde{T}_i \geq t \,|\, X_{i,[0,t_{\mathrm{P}}]}, \tilde{T}_i > t_{\mathrm{P}}; \hat{\theta}), \tag{2.15}$$

where $\hat{\theta}$ is the optimal parameter of the trained model. Figure 2.5 below shows individuals in the training set and the prediction scenario of individuals in the testing set.

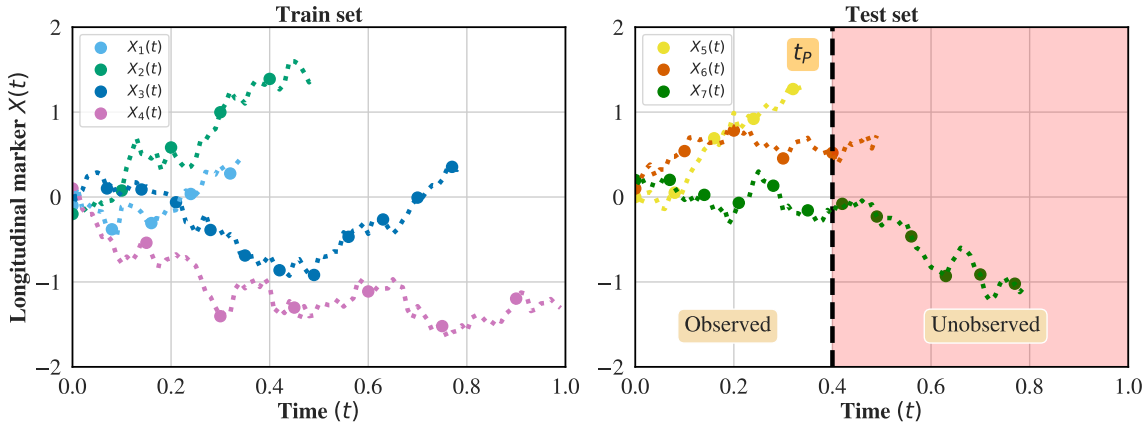

Figure 2.5 – The train set includes 4 individuals $i \in \{1, 2, 3, 4\}$ (**right**) and the test set 3 individuals $i \in \{5, 6, 7\}$ (**left**). In particular, on the **right**, we are interested in predicting the survival probability of individuals in the test set that are alive at time $t_{\mathrm{P}} = 0.4$. In this case, we ignore the individual $i = 5$ whose survival time is less than 0.4 and predict the survival probability of individual $i \in \{6, 7\}$ with the set of observed longitudinal markers up to time t_{P} (the longitudinal markers after time t_{P} are considered as unobserved).

Dynamic prediction with longitudinal data. In the context of longitudinal data which is measured repeatedly over time, the prediction could be dynamically updated as additional longitudinal information becomes available. Figure 2.6 shows the dynamic prediction of the conditional survival probability for individuals in the testing set.

To effectively predict the survival probability defined in (2.15), there are two widely used approaches which are landmarking and joint modeling. The landmarking approach

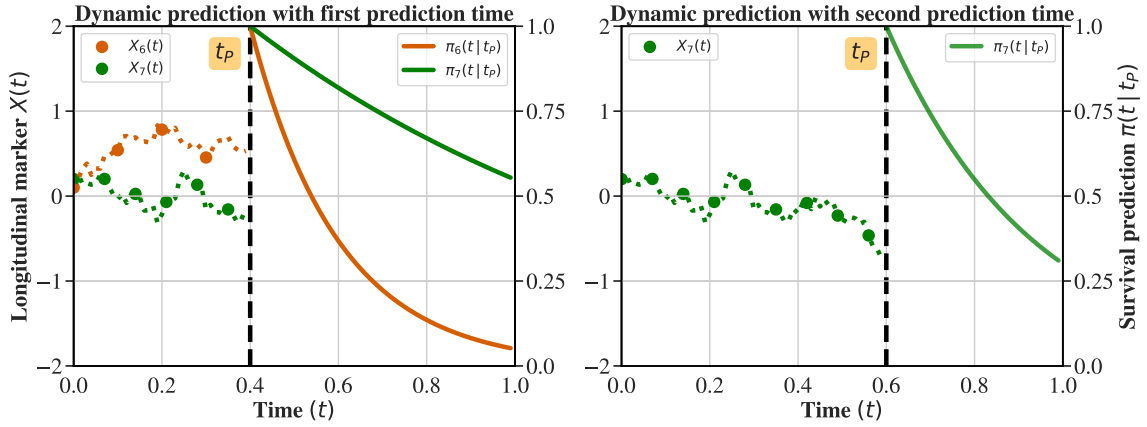

Figure 2.6 – The dynamic prediction of conditional survival probability $\pi(t|t_{\mathrm{P}})$ at two different prediction times $t_{\mathrm{P}} = 0.4$ (**left**) and $t_{\mathrm{P}} = 0.6$ (**right**) for individuals who survive at corresponding prediction time, given the historical longitudinal marker up to that time.

fits a survival model only to a subset of data consisting of individuals still at risk at the prediction time and the historical longitudinal marker up to the prediction time. In contrast, the joint model uses the whole dataset to build concurrently a longitudinal model for longitudinal data and a survival model for time-to-event data while taking into account the dependency between these two models. In the following, we present the framework of these two approaches in detail.

2.2.2 Landmark methods

The landmarking approach (Anderson et al., 1983; Van Houwelingen, 2007; Devaux et al., 2022) provides an estimation of $\pi_i(t \mid t_{\mathrm{P}})$ by building a survival model on a set $R(t_{\mathrm{P}}) = \{i : T_i \geq t_{\mathrm{P}}\}$, which includes individuals from the training set who have survived up to the time point t_{P}, known as the landmark time. It is important to note that the chosen landmark time coincides with the prediction time. For these individuals, the model incorporates their longitudinal markers recorded up to landmark time, denoted by $X_{i,[0,t_{\mathrm{P}}]}$. A summary of the longitudinal markers $X_{i,[0,t_{\mathrm{P}}]}$, denoted by $\phi(X_{i,[0,t_{\mathrm{P}}]})$, is derived and can take the form of the last observed value (Proust-Lima and Taylor, 2009; Sweeting et al., 2017) or other functionals that capture the history of longitudinal features (Devaux et al., 2022). This summary is then used as the baseline features in the survival model (see Section 2.2.4 for more details).

Any survival prediction model that uses $\phi(X_{i,[0,t_{\mathrm{P}}]})$ as baseline features, such as Cox proportional hazards or random survival forest, can be applied within this framework. Figure 2.7 below shows an illustration of training and prediction in this landmark setting.

For example, if a Cox-PH model is used, the hazard function defined for all $t \geq t_{\mathrm{P}}$ as

$$\lambda(t \mid X_{i,[0,t_{\mathrm{P}}]}) = \lambda_0(t \mid t_{\mathrm{P}}) \exp(\phi(X_{i,[0,t_{\mathrm{P}}]})^\top \alpha(t_{\mathrm{P}})).$$

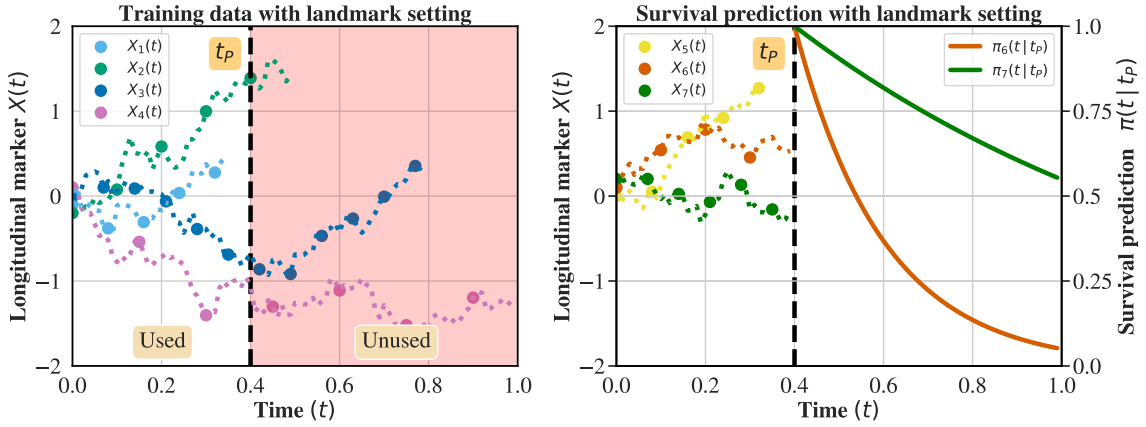

Figure 2.7 – We set the landmark time to 0.4. On the **left**, we select individuals whose survival times are greater than the landmark time. In this case $R(t_{\mathrm{P}}) = \{2, 3, 4\}$ and we ignore the individual $i = 1$. The longitudinal markers observed only up to the landmark time are used to train the survival model while data beyond the landmark time are ignored. On the **right**, we predict the conditional survival probability $\pi(t|t_{\mathrm{P}})$ of individuals who survive at landmark time with their longitudinal markers observed up to landmark time. In this case, we ignore the individual $i = 5$ and derive the simulated survival probability prediction of individual $i \in \{6, 7\}$.

The estimation of the vector $\alpha(t_{\mathrm{P}})$ and of the function $\Lambda_0(. \mid t_{\mathrm{P}})$ can be obtained by maximizing the log-likelihood similarly defined in (2.12). The estimated cumulative baseline hazard function, which is also similar to (2.11), takes the form

$$\hat{\Lambda}_0(t \mid t_{\mathrm{P}}) = \sum_{i \in R(t_{\mathrm{P}})} \frac{\mathbb{1}_{T_i \leq t} \Delta_i}{\sum_{j:T_j \geq T_i} \exp(\phi(X_{j,[0,t_{\mathrm{P}}]})^\top \hat{\alpha}(t_{\mathrm{P}}))},$$

where $\hat{\alpha}(t_{\mathrm{P}})$ is an estimator of $\alpha(t_{\mathrm{P}})$. The predicted conditional survival function for a new individual i, which is similar to (2.13), is then given by

$$\pi_i(t \mid t_{\mathrm{P}}) = \exp\Big(- \hat{\Lambda}_0(t \mid t_{\mathrm{P}}) \exp\left(\phi(X_{i,[0,t_{\mathrm{P}}]})^\top \hat{\alpha}(t_{\mathrm{P}})\right)\Big).$$

The landmarking approach offers the benefit of being computationally simpler than joint modeling, making it easier to scale for high-dimensional problems. However, its drawbacks include inefficiency due to training on only a subset of the dataset, specifically by excluding individuals whose survival time is shorter than the landmark time and ignoring the longitudinal markers collected after that time (Lee et al., 2019; Devaux et al., 2022). In addition, to obtain predictions at a new landmark time, the model needs to be entirely retrained with the new subset of at-risk individuals and the historical longitudinal marker up to the new landmark time.

2.2.3 Joint models

The joint modeling approach (Wulfsohn and Tsiatis, 1997; Lin et al., 2002b; Proust-Lima et al., 2009; Proust-Lima and Taylor, 2009; Andrinopoulou and Rizopoulos, 2016) simultaneously analyses longitudinal and time-to-event data, taking into account the potential dependence between them. More specifically, it consists of defining:

- a survival model for modeling the time-to-event data, often by a Cox-PH model;
- a longitudinal model for modeling the longitudinal markers. It could be a (generalized) linear mixed-effects model or another appropriate longitudinal model depending on the nature of the data (continuous, binary, counts, etc);
- a linking mechanism for linking both models via a common latent structure.

We now present the most common models used for modeling the longitudinal markers, which is the linear mixed model, and then two main linking mechanisms, which are the shared parameter joint model and the joint latent class model.

Linear mixed model

Although longitudinal data from individuals are assumed to be independent, they often show correlations between repeated measurements within the same individual over time. Linear mixed models (LMM) (Laird and Ware, 1982) are commonly used to analyze longitudinal data by including both fixed and random effects. The fixed effect, which is common to all individuals, describes the average longitudinal evolution in time. The marker-specific random effect, which is unique to each individual, describes how each individual deviates from the average evolution. Moreover, this random effect is the key factor that encodes the correlation between the different longitudinal markers.

In LMM, for each individual $i \in \{1, \ldots, n\}$ and its univariate longitudinal feature $\ell \in \{1, \ldots, d\}$, the observation at time $t \in \{t_i^1, \ldots, t_i^{n_i}\}$ is assumed to be

$$X_{i\ell}(t) = x_{i\ell}(t) + v(t)^\top b_{i\ell} + \epsilon_{i\ell}^j, \tag{2.16}$$

where the error term $\epsilon_{i\ell}^j$ is assumed to be normally distributed, $\epsilon_{i\ell}^j \sim \mathcal{N}(0, \sigma_\ell)$ with $\sigma_\ell \in \mathbb{R}^+$ being an estimated standard deviation parameter. The term $v_\ell(t) \in \mathbb{R}^{r_\ell}$ is a row vector of time-varying features—often including functions of time, such as linear slopes, cubic splines, or polynomial terms (see Rizopoulos, 2012)—associated with an unknown random effect $b_{i\ell} \in \mathbb{R}^{r_\ell}$. In addition, $x_{i\ell}(t) = u(t)^\top \beta_\ell$ represents the fixed effects component, where $u_\ell(t) \in \mathbb{R}^{q_\ell}$ is another row vector of time-varying features similar to $v_\ell(t)$, with corresponding unknown fixed effect parameters $\beta_\ell \in \mathbb{R}^{q_\ell}$. The random effect $b_{i\ell}$ is typically assumed to follow a zero-mean multivariate normal distribution, $b_{i\ell} \sim \mathcal{N}(0, D_{\ell\ell})$, where $D_{\ell\ell} \in \mathbb{R}^{r_\ell \times r_\ell}$ is the covariance matrix.

Dependency between longitudinal features. In addition to the correlation between repeated measurements of longitudinal data within the same individual over time, there is also the correlation between the multiple longitudinal features in multivariate problems.

This correlation can be modeled through the random effect (Xu and Zeger, 2001) or the error term (Chi and Ibrahim, 2006). For each individual $i \in \{1, \ldots, n\}$, we let $b_i = (b_{i1}, \ldots, b_{id})$ be the stacked vector of random effects for all d longitudinal features and $\epsilon_i^j = (\epsilon_{i1}^j, \ldots, \epsilon_{id}^j)$ be the stacked vector of error terms for all d longitudinal features. If the random effect is used to account for the correlation between longitudinal features, the random effect and error term are respectively

$$b_i \sim \mathcal{N}(0, D) \quad \text{and} \quad \epsilon_{i\ell}^j \sim \mathcal{N}(0, \sigma_\ell),$$

where D is a variance-covariance matrix that captures both the correlation between longitudinal features and repeated measures. On the other hand, if the error term is used to account for the correlation between longitudinal features, the random effect and error term are respectively

$$b_{i\ell} \sim \mathcal{N}(0, D_{\ell\ell}) \quad \text{and} \quad \epsilon_i^j \sim \mathcal{N}(0, \Sigma),$$

where Σ is a variance-covariance matrix that captures the dependency between longitudinal features measured at the same time. Other distributions have been studied, see Hickey et al. (2016).

In addition, we can use a generalized linear mixed model (Hickey et al., 2016), which is an extension of the LMM, to analyze other types of longitudinal data such as binary, counts.

Linking mechanism. The association between the longitudinal features and survival time can be described by different linking mechanisms. Two key approaches for modeling this relationship are the shared parameter joint model (SREM) and the joint latent class model (JLCM). Figure 2.8 below gives a graphical representation illustrating the dependence structure for these two linking mechanisms.

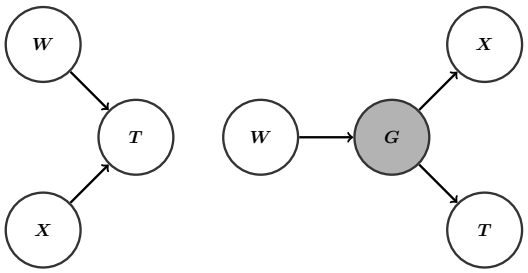

Figure 2.8 – Graphical representation of SREM (**left**) and JLCM (**right**). The variable W represents time-independent features, X the longitudinal markers, T the time-to-event, and G the latent class membership.

We now provide a detailed description of these two linking mechanisms. It is important to note that additional models are also discussed in the literature, see Hickey et al. (2016) or Rizopoulos (2012) for a more complete review.

Shared parameter joint model

The shared parameter joint model (SREM) (Wulfsohn and Tsiatis, 1997; Andrinopoulou and Rizopoulos, 2016) assumes that the dependence between the longitudinal marker and the time-to-event within individuals is captured by the random effect, which represents individual-specific variability. The random effect is then included as a covariate in both the longitudinal and the survival models. For an individual i, if we choose the LMM - defined in (2.16) - for the longitudinal markers and Cox-PH model for the time-to-event, the hazard function which depends on the d longitudinal features assuming for each of them H functional forms, can be written as

$$\lambda(t \mid W_i, b_i) = \lambda_0(t) \exp\left(W_i^\top \alpha + \phi(x_{i,[0,t]})^\top \eta + \sum_{\ell=1}^{d} \sum_{h=1}^{H} \psi_{\ell h}(b_{i\ell}, t)^\top \gamma_{\ell h}\right),$$

where W_i are the baseline features with the corresponding coefficients α, and $\psi_{\ell h}$ is the shared association function with its associated parameter γ. The function ψ can take different forms such as random effect, generalized random effects, and fixed effects (Hickey et al., 2016), providing flexibility in capturing how the random effect from the longitudinal model influences survival times. In addition, the dependence between the longitudinal features and survival time can be described by the term $\phi(x_{i,[0,t]})$ in which $x_{i,[0,t]}$ is the true unobserved longitudinal markers estimated from generalized linear mixed model and ϕ can be in different forms such as current value, current slope, area under the curve (Andrinopoulou and Rizopoulos, 2016; Hickey et al., 2016).

This linking mechanism allows the longitudinal markers to influence the risk of the event in the survival model while taking into account the correlation between the two types of data. The likelihood function is the joint likelihood of the longitudinal model and the survival model which requires an integration over the random effect distribution. The advantage of this approach is that it allows flexibility in modeling the association between longitudinal data and survival data through different types of function ψ. However, a significant disadvantage is that it requires a lot of computational power due to the numerical integration over the random effect distribution in the likelihood function (Proust-Lima et al., 2014). This disadvantage of the SREM approach can be mitigated by another linking mechanism which is JLCM.

Joint latent class model

The joint latent class model (JLCM) (Lin et al., 2002b; Proust-Lima et al., 2009; Proust-Lima and Taylor, 2009) assumes that the population is heterogeneous, comprising $K \in \mathbb{N}^*$ latent classes. Within each homogeneous latent class $g \in \{1, \ldots, K\}$, individuals share similar marker trajectories and have the same risk of the event. For each $k \in \{1, \ldots, K\}$, the latent class membership probability for an individual i is assumed to take the form of

multinomial logistic regression (Böhning, 1992), that is,

$$\mathbb{P}[g_i = k \mid W_i] = \frac{e^{W_i^\top \xi_k}}{\sum_{j=1}^{K} e^{W_i^\top \xi_j}},$$

where $\xi_k \in \mathbb{R}^p$ is a vector of coefficients for the class k associated to the baseline features W_i. The dependence between the time-to-event and the longitudinal marker is fully captured by this latent class structure, which also means there are no shared associations between the longitudinal model and the survival model. Given the latent class membership, these two models are assumed to be independent. For an individual i with given latent class $g_i = k$, if we choose the LMM - defined in (2.16) - for longitudinal marker $\ell \in \{1, \dots, d\}$ and the Cox-PH model for the time-to-event (Proust-Lima et al., 2014), we have, in the simplest form,

$$\begin{aligned} X_{i\ell}(t \mid g_i = k) &= u(t)^\top \beta_{\ell k} + v(t)^\top b_{i\ell} + \epsilon_{i\ell}^j, \\ \lambda(t \mid W_i, g_i = k) &= \lambda_0(t) \exp\Big(W_i^\top \alpha_k + \phi(x_{i,[0,t]})^\top \eta_k + \psi(t)^\top \gamma_k\Big), \end{aligned}$$

where $\beta_{\ell k}$ is the fixed effect parameter for class k in the longitudinal model for marker ℓ, and α_k, η_k and γ_k are vectors of coefficients for class k associated respectively with the baseline features W_i, the association function of the true unobserved longitudinal markers $\phi(x_{i,[0,t]})$ and the association function of time $\psi(t)$ in the survival model.

The advantage of JLCM is that it makes less assumptions about the dependence on the time-to-event and longitudinal markers as well as offers a computationally attractive alternative to SREM. However, a drawback is that it requires more parameters when the number of latent classes is large, and it is more difficult to interpret the latent classes and the parameters within each class than SREM (Proust-Lima et al., 2014).

Figure 2.9 shows the two fitted submodels of the SREM with their estimated longitudinal marker and hazard function on the training set.

Survival prediction with joint model. Once all the parameters in the joint model are estimated, for a new individual j with its historical longitudinal markers up to prediction time t_{P}, the prediction of the longitudinal marker denoted by $\hat{X}_j(t|t_{\mathrm{P}})$ and the conditional survival function $\pi_j(t|t_{\mathrm{P}})$ at all time $t \geq t_{\mathrm{P}}$ can be both computed (Proust-Lima et al., 2014). Figure 2.10 below shows these simulated predictions for individuals on the testing set.

Overall, the advantage of the joint modeling approach is that it makes more efficient use of the data by incorporating all longitudinal information into a survival model (Yu et al., 2004) compared with the landmark approach, which uses only information from individuals at risk at the landmark time (Devaux et al., 2022) and the model is trained only once on the entire training dataset and can be used to update the prediction as the new information is recorded. The disadvantage of this approach is that it relies on strong assumptions about the relationship between longitudinal data and survival outcomes, is computationally expensive, and does not scale to high-dimensional problems with a large number of longitudinal markers (Devaux et al., 2022).

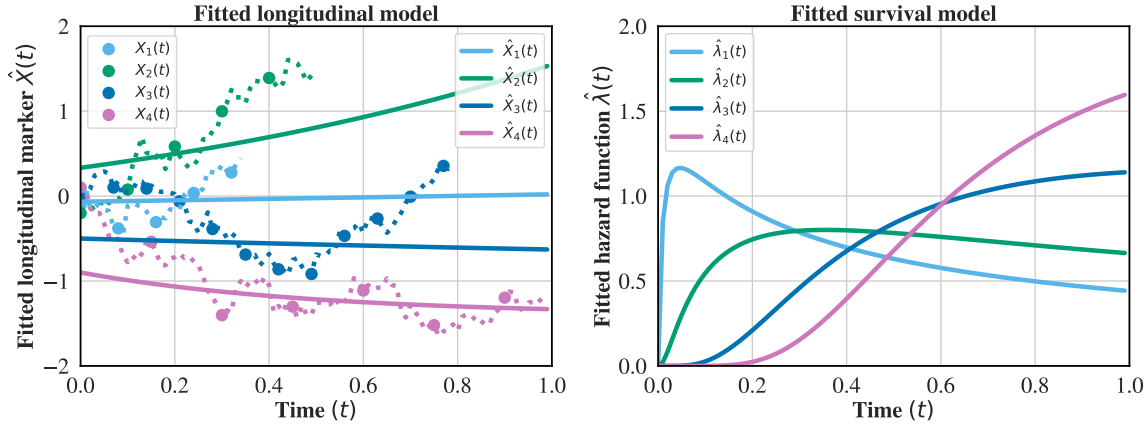

Figure 2.9 – The two submodels of the SREM fit on the training set: longitudinal model (i.e. LMM) on the **left**, survival model (i.e. Cox-PH) on the **right**. In particular, for individuals on the training set, the estimated longitudinal markers over time t, denoted by $\hat{X}(t)$, are derived from the fitted longitudinal model (the straight lines on the **left**), and the estimated hazard function, denoted by $\hat{\lambda}(t)$, is derived from the fitted survival model (on the **right**).

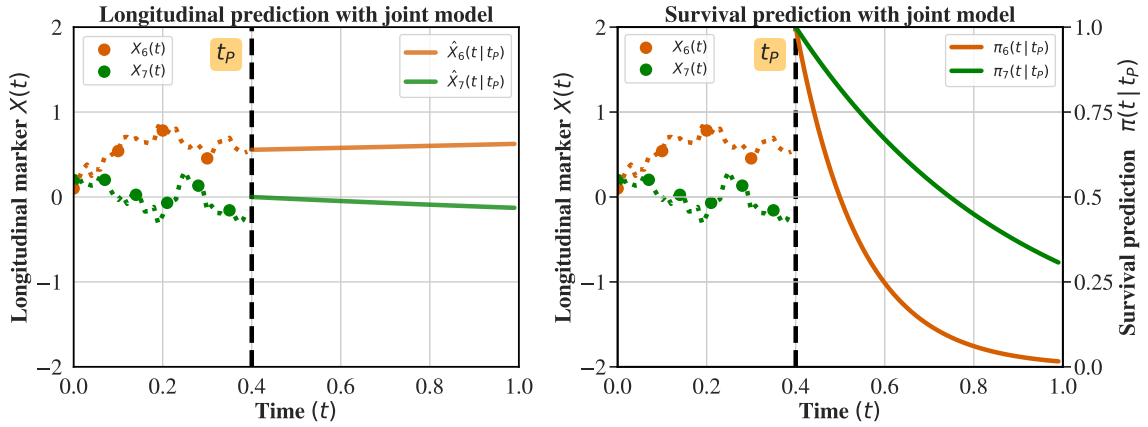

Figure 2.10 – The prediction of longitudinal marker $\hat{X}(t|t_{\mathrm{P}})$ (**left**), conditional survival probability $\pi(t|t_{\mathrm{P}})$ (**right**) for individuals who survive at prediction time t_{P}, given the historical longitudinal marker up to t_{P}.

2.2.4 Featuring for longitudinal markers

The association function $\phi(x_{i,[0,t]})$, which summarizes the historical longitudinal features up to time t to model their effect on survival risk, is common in both landmark and joint model approaches. It can take various forms. Selecting the appropriate form is crucial, as it can significantly influence the results (Andrinopoulou and Rizopoulos, 2016). Both approaches consider different functional forms (Andrinopoulou and Rizopoulos, 2016; Hickey et al., 2016; Devaux et al., 2022) such as

- current value

$$\phi(x_{i,[0,t]}) = x_i(t),$$

- current slope
$$\phi(x_{i,[0,t]}) = \frac{dx_i(t)}{dt},$$
- area under the curve
$$\phi(x_{i,[0,t]}) = \int_0^t x_i(u)du.$$

tsfresh. In addition, a comprehensive number of features to summarize $\phi(x_{i,[0,t]})$, including statistical measures, time-domain characteristics, and frequency-domain transformations, can be automatically calculated using the *tsfresh* package (Christ et al., 2018). This Python tool, designed for time series feature extraction, is also effective at extracting complex patterns from multiple longitudinal markers simultaneously. The utility and efficiency of this package have been recognized and used in various research studies (Yang et al., 2021; Santis et al., 2022), proving it to be an essential resource in time series analysis.

Splines. Other featuring methods or definitions of the association function can be used. In the literature, the most common are the splines, which generate a smooth curve to represent observed longitudinal markers. The spline-fitting process divides the time domain into intervals, determined by knots, which specify where the data is segmented. Within each interval, a polynomial function is fitted to the data. To ensure smooth transitions, splines are constrained at the knots, ensuring continuity in both value and slope across the segments. A widely used approach is the cubic spline, where a spline with K knots can be modeled as follows:

$$\phi(x_{i,[0,t]}) = \nu_0 + \nu_1 B_1(t) + \nu_2 B_2(t) + \ldots + \nu_{K+3} B_{K+3}(t),$$

where $B_1, B_2, \ldots, B_{K+3}$ are basis functions, $\nu_0, \nu_1, \nu_2, \ldots, \nu_{K+3}$ are the coefficients of the spline and can be estimated using ordinary least squares (Gareth et al., 2013).

While splines can efficiently fit a model for each longitudinal marker (De Boor, 1978), they can also be extended to fit multiple longitudinal markers simultaneously by using shared parameters across markers (Wood, 2017). This capability allows splines to handle more complex data structures where relationships between markers are important.

Signatures. The summarizing $\phi(x_{i,[0,t]})$ can be obtained by a signature transformation (Lyons et al., 2007; Friz and Victoir, 2010; Fermanian, 2021). This method has its origins in stochastic analysis, first introduced by Chen (1958) and later developed by Lyons et al. (2007). Recently, they have been successfully applied in statistics and machine learning as a powerful tool for representing irregular time series data (Kidger et al., 2019; Bleistein et al., 2023; Fermanian, 2022). Mathematically, the signature associated to word $I = (\ell_1, \ldots, \ell_k) \in \{1, \ldots, d\}^k$ of size k over a set of longitudinal markers $x_{i,[0,t]}$ is defined as the mapping

$$t \mapsto \mathbf{S}^I(x_{i,[0,t]}) := \int_{0<u_1<\cdots<u_k<t} dx_{i\ell_1}(u_1) \ldots dx_{i\ell_k}(u_k).$$

While the technical definition of the signature involves iterated integrals, it can be viewed more intuitively as a feature extraction technique that captures important characteristics of the time series. Figure 2.11 below shows the example of signature transformation on a set of longitudinal markers.

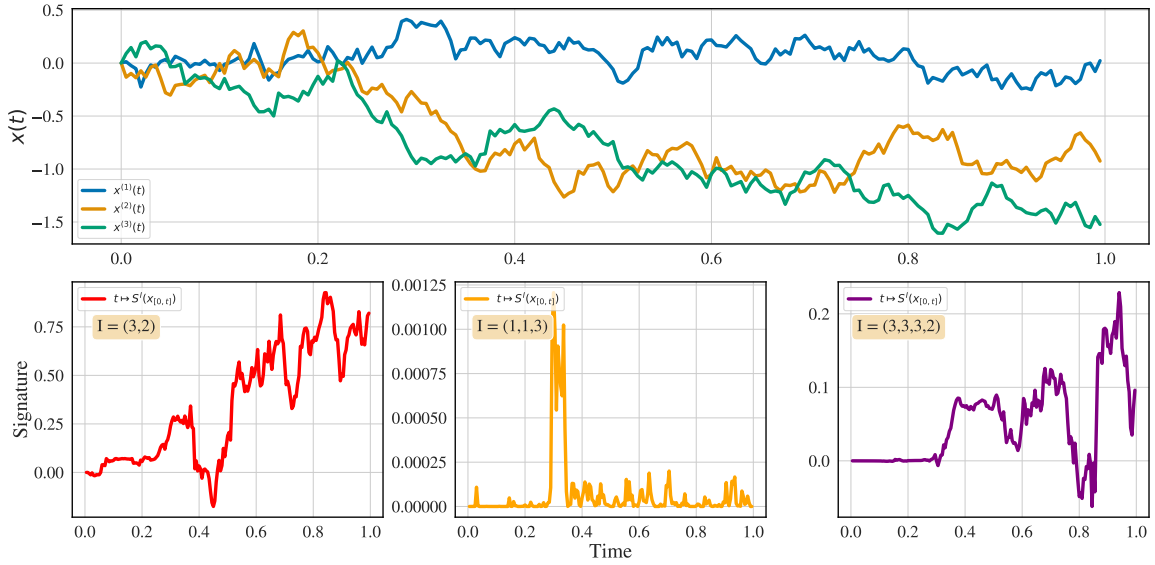

Figure 2.11 – A 3-dimensional longitudinal features $x(t)$ on **top**, and three signature coefficients $\mathbf{S}^I(x_{[0,t]})$ associated to different words on the **bottom**.

The truncated signature of order $N \geq 1$, denoted as $\mathbf{S}_N(x_{[0,t]})$, consists of all the signature coefficients corresponding to words of size $k \leq N$, arranged in lexicographical order. This can be expressed as

$$\mathbf{S}_N(x_{i,[0,t]}) = \left(\mathbf{S}^I(x_{i,[0,t]})\right)_{|I|\leq N}.$$

The order N is an important hyperparameter in the model, controlling the complexity and capturing higher-order interactions in the time series. To further illustrate how signature transformation encodes geometric properties and captures interactions between different coordinates, Figure 2.12 shows the signature coefficients extracted from two longitudinal markers that are mostly similar over time.

Although these two markers remain nearly identical for most of the time, the differences that occur at the initial stages are still captured and encoded in the extracted signature coefficients, demonstrating the ability of the signature transformation to retain early variations over time.

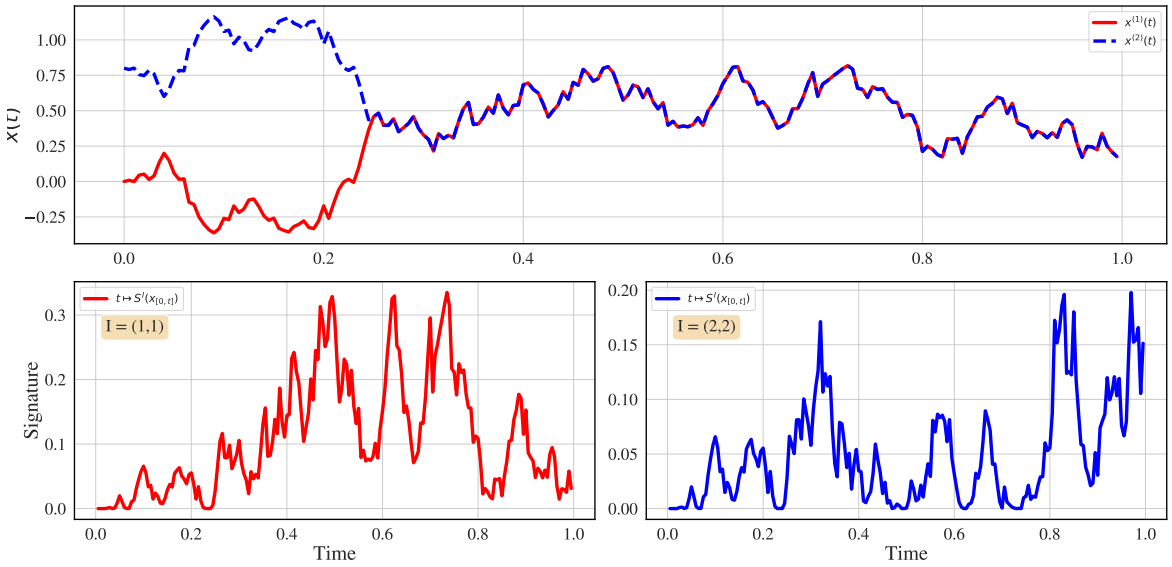

Figure 2.12 – A 2-dimensional longitudinal features $x(t)$ on **top**, in which both features appear similar after time $t = 0.23$, and two signature coefficients $\mathbf{S}^I(x_{[0,t]})$ on the **bottom** associated to word $(1, 1)$ (**left**) and $(2, 2)$ (**right**) respectively.

2.3 Deep survival methods with longitudinal data

In addition to deep neural network frameworks that handle baseline features, described in Section 2.1.5, a variety of frameworks have been developed to incorporate longitudinal data (Lee et al., 2019; Gupta et al., 2019; Avati et al., 2020; Groha et al., 2020; Nagpal et al., 2021; Moon et al., 2022). Most of these frameworks use Recurrent neural network (RNN) architectures to handle longitudinal features, which are suitable for taking into account temporal information and sharing parameters across time (Wiegrebe et al., 2024).

Gupta et al. (2019), Avati et al. (2020), Groha et al. (2020), and Nagpal et al. (2021) also propose frameworks that use standard RNN architecture to learn the representations of the longitudinal data and then feed to model the survival data in different settings (single risk, competing risk, ...). We focus in particular on *Dynamic DeepHit* (Lee et al., 2019) which is a state-of-the-art method for dynamical survival analysis.

Dynamic DeepHit (DDH) follows a similar structure to *DeepHit*, which uses the discrete-time model (discussed in Section 2.1.4) and consists of two subnetworks. The first subnetwork is an RNN that embeds the longitudinal markers X into hidden states h, which are then forwarded through fully connected layers (FF) to derive the estimation of longitudinal markers $\hat{X}$. For each individual $i \in \{1, \dots, n\}$ with the historical longitudinal markers $X_{i,[0,\tau_j]}$ up to each discrete time $\tau_j \in \{\tau_1, \dots, \tau_J\}$, the hidden state h_{ij} and the estimation of longitudinal markers $\hat{X}_i(\tau_j)$ are sequentially updated as

$$h_{ij} = \text{RNN}(h_{i(j-1)}, X_i(\tau_j), m_{ij}) \qquad \text{and} \qquad \hat{X}_i(\tau_j) = \text{FF}(h_{ij}),$$

where m_{ij} denotes an indicator for missing measurements, as longitudinal markers may not

always be observed at every time point. The hidden states $\{h_{ij}\}_{j=1}^{J}$ are forwarded through an attention mechanism, which helps the network decide which part of the history of the measurements is important and derive a context vector c_i as a weighted sum of the previous hidden states

$$c_i = \sum_{j=1}^{J} a_{ij} h_{ij},$$

where $\{a_{ij}\}_{j=1}^{J}$ represents the importance of the measurements at time τ_j. The second subnetwork takes the context vector c as input and derives the estimation of the probability mass function at its output layer, which is similar to *DeepHit*, described in Section 2.1.5. A key limitation in this framework arises when the probability mass function at times prior to the observed time is estimated using the entire longitudinal marker up to this observed time. This incorporates future information into the model, potentially reducing the accuracy of dynamic predictions by including data that would not be accessible in a real-time setting. Consequently, this data leakage reduces the reliability of the model for real-time prediction. To provide a clearer understanding of the framework, Figure 2.13 offers a illustration of the architecture.

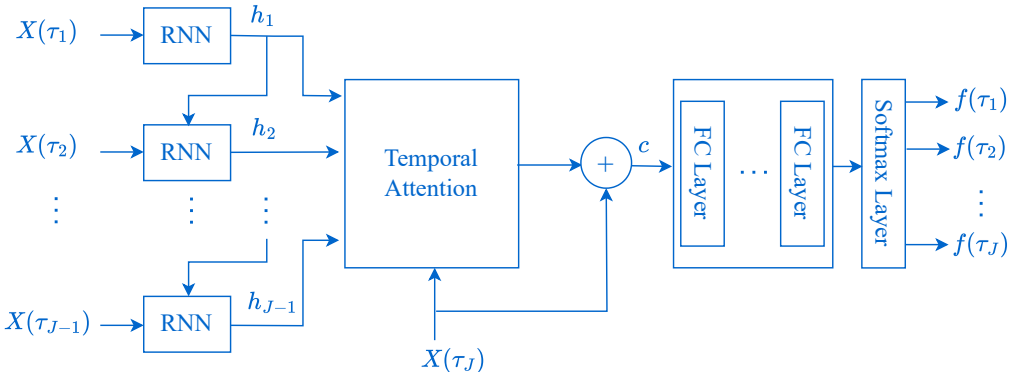

Figure 2.13 – A simplified illustration of Dynamic DeepHit processing longitudinal data X, measured at discrete time points $\{\tau_1, \ldots, \tau_J\}$, to estimate the probability mass functions f at each of these time points.

The loss function $\mathcal{L}^{\text{DDH}}(\theta)$ is a sum of three loss functions, which is defined as

$$\begin{aligned}
&\mathcal{L}^{\text{DDH}}(\theta) \\
&= \sum_{i=1}^{n} \left[(1 - \Delta_i) \log \left(S_\theta(T_i \mid X_i) \right) + \Delta_i \log \left(\frac{f_\theta(T_i \mid X_i)}{1 - \sum_{\tau_{j'} \le t_i^{n_i}} f_\theta(\tau_{j'} \mid X_i)} \right) \right] \\
&+ \alpha \sum_{i=1}^{n} \sum_{\substack{j \neq i \\ j=1}}^{n} \eta \mathbb{1}_{T_i - t_i^{n_i} < T_j - t_j^{n_j}} \Delta_i \big(F_\theta(T_i \mid X_i), F_\theta(T_i - t_i^{n_i} + t_j^{n_j} \mid X_j) \big) \\
&+ \beta \sum_{i=1}^{n} \sum_{j=1}^{J} \sum_{\ell=1}^{L} (1 - m_{ij\ell}) \xi(X_{i\ell}(\tau_j) - \hat{X}_{i\ell}(\tau_j)),
\end{aligned}$$

where α and β are hyper-parameters, and $\hat{X}$ is the estimation of X. The first and second losses are similar to the ones of *DeepHit*, which are the log-likelihood and the ranking loss respectively. The third loss is a prediction loss, which measures the difference between the value of the time-dependent features and a prediction of this value made by the shared network. The loss is minimized using Adam (Kingma and Ba, 2014). We refer the reader to Lee et al. (2019) for a more detailed description.

Moon et al. (2022) proposes *SurvLatent ODE*, which extends *Latent ODE* (Rubanova et al., 2019) to handle survival data. This framework is based on a variational autoencoder architecture (Kingma and Welling, 2013), which consists of a recognition model and a generative model. The recognition model handles the longitudinal features sequentially backward in time and outputs the posterior over the initial latent state. The generative model uses a sample from the posterior over the initial latent state to derive the latent states at all measurement times and then to estimate the distribution of survival time. In this framework, the encoder follows the ODE-RNN architecture (Rubanova et al., 2019), where an ODE solution is first computed as a pre-update term, and this is passed through an RNN to iteratively update the hidden state at each measurement time. The ODE-RNN encoder runs backwards-in-time from τ_J to τ_1. For an individual i with longitudinal marker X_i, the hidden state $h_i(\tau_j)$ at each measurement time $\tau_j \in \{\tau_J, \ldots, \tau_1\}$ is sequentially updated as

$$
\begin{aligned}
h_i'(\tau_j) &= h_i(\tau_{j+1}) + \int_{\tau_j}^{\tau_J} f_\gamma(h_i(u), u)du, \\
h_i(\tau_j) &= \text{RNN}(h_i'(\tau_j), X_i(\tau_j)),
\end{aligned}
$$

where $h_i(\tau_J)$ is an initial hidden state and the function $f_\gamma(.)$, parameterized by neural networks with weights γ, specifies the dynamics of RNN hidden states. The final hidden state of the recognition model ODE-RNN $h_i(\tau_1)$ is then forwarded to a neural network, whose weight is denoted by ϕ, to estimate the mean and variance of the posterior over the initial latent state $q(z_i(\tau_1)|X_i)$, that is

$$
q(z_i(\tau_1)|X_i) = \mathcal{N}(\mu_{z_i(\tau_1)}, \sigma_{z_i(\tau_1)}) \quad \text{where} \quad \mu_{z_i(\tau_1)}, \sigma_{z_i(\tau_1)} = g_\phi(h_i(\tau_1)).
$$

The generative model consists of an ODE and a fully connected layers architecture network, in which the latent trajectory derived from the ODE model is forwarded into the subnetwork model to estimate the discrete hazard functions. The latent state at any point $\tau_j \in \{\tau_1, \ldots, \tau_J\}$ is the solution to the ODE

$$
z_i(\tau_j) = z_i(\tau_1) + \int_{\tau_1}^{\tau_j} f_\omega(z_i(u), u)du,
$$

where $z_i(\tau_1)$ is a sample from the approximate posterior $q(z_i(\tau_1)|X_i)$,and f_ω, a neural network parameterized by weights ω, specifies the dynamics of the latent state. The subnetwork then takes all latent states $\{z(\tau_1), \ldots, z(\tau_J)\}$ as input and derives the discrete hazard function at all times $\{\lambda(\tau_1), \ldots, \lambda(\tau_J)\}$.

The loss function, which is the combination of the log-likelihood and the Kullback-Leibler divergence loss, is defined as

$$\begin{aligned}
&\mathcal{L}^{\text{SLODE}}(\theta) \\
&= \sum_{i=1}^{n} \Delta_i \log\big(\lambda_\theta(T_i \mid X_i)\big) + (1-\Delta_i)\log\big(1-\lambda_\theta(T_i \mid X_i)\big) + \sum_{j:\tau_j<T_i} \log\big(1-\lambda_\theta(\tau_j \mid X_i)\big) \\
&+ \mathbb{E}_{q(z_i(\tau_1)\mid X_i)}\Big[\log\big(p(X_i \mid z_i(\tau_1))\big] - \text{KL}[q\big(z_i(\tau_1) \mid X_i\big) \,||\, p\big(z_i(\tau_1)\big)\Big].
\end{aligned}$$

We refer the reader to Moon et al. (2022) for a more detailed description.

2.4 Evaluation of survival analysis

The predictive performance of a survival model is the degree to which the predicted distribution of survival time matches with the observed times. In the literature, this predictive performance is assessed by discrimination (Harrell Jr et al., 1996; Gönen and Heller, 2005), or calibration (Schemper and Henderson, 2000; Gerds and Schumacher, 2006). Discrimination describes the ability of the model to distinguish between individuals whose event occurs earlier (shorter survival time) and those whose event occurs later (longer survival time). Calibration, on the other hand, describes the similarity between the predicted probabilities and the observed probabilities (derived from the observed time) on the same individual. In this section, we present different metrics of discrimination and calibration adapted to the dynamic prediction context, where the predicted probability is updated as new information of longitudinal marker is available. All these metrics are functions of the predicted survival probabilities defined in (2.15).

Discrimination

In the context of dynamic prediction, we focus on a time interval within which the occurrence of events is of interest. In this context, a useful property of the model would be to successfully discriminate between individuals who are going to experience the event within this time frame from individuals who will not. More formally, given longitudinal marker up to prediction time t_{P}, we are interested in events occurring in the time interval $[t_{\text{P}},\, t]$.

Time-dependant Concordance index (C-Index) is the most commonly used metric for discrimination in survival analysis. This metric can be computed as the proportion of concordant pairs over comparable pairs. In a dynamic prediction context (Lee et al., 2019), for a randomly chosen pair of individuals (i, j), in which both individuals have provided measurements up to the time t_{P}, (i, j) are comparable if $T_i > T_j$, $T_j \in [t_{\text{P}},\, t]$ and $\Delta_j = 1$. This pair (i, j) is concordant if $\pi_j(t \mid t_{\text{P}}) < \pi_i(t \mid t_{\text{P}})$. The concordance index $C(t_{\text{P}}, t)$ is then

defined as

$$\frac{\sum_{j=1}^{n}\sum_{i=1}^{n}\mathbb{1}_{\pi_i(t\,|\,t_\mathrm{P})>\pi_j(t\,|\,t_\mathrm{P})}\mathbb{1}_{T_i>T_j,\,T_j\in[t_\mathrm{P},\,t],\Delta_j=1}}{\sum_{j=1}^{n}\sum_{i=1}^{n}\mathbb{1}_{T_i>T_j,\,T_j\in[t_\mathrm{P},\,t],\Delta_j=1}}.$$

Since the C-index is a proportion, it can take any value from 0 to 1. Values near 1 indicate high performance and a value of 0.5 indicates that the discrimination performance of the model is the same as a random guess.

Receiver Operating Curve (or ROC) is another metric to distinguish between individuals who will have the event occur within the considered time frame from individuals who will not. This metric is calculated through two terms, i.e., specificity and sensitivity. The sensitivity (or true positive rate) is the proportion between individuals predicted to have the event occur in the time interval $[t_\mathrm{P},\,t]$ and the individuals whose true survival times are in the time interval. The specificity (or true negative rate) is the proportion of individuals predicted to have the event occur beyond the time interval $[t_\mathrm{P},\,t]$ over the individuals whose true survival times are after this time interval. These two terms are then in form

$$\mathrm{TPR}_{[t_\mathrm{P},\,t]}(c) = \frac{\sum_{i=1}^{n}\mathbb{1}_{\pi_i(t\,|\,t_\mathrm{P})>c}\mathbb{1}_{T_i\in[t_\mathrm{P},\,t],\Delta_j=1}}{\sum_{i=1}^{n}\mathbb{1}_{T_i\in[t_\mathrm{P},\,t],\Delta_j=1}},$$

and

$$\mathrm{TNR}_{[t_\mathrm{P},\,t]}(c) = \frac{\sum_{i=1}^{n}\mathbb{1}_{\pi_i(t\,|\,t_\mathrm{P})<c}\mathbb{1}_{T_i>t}}{\sum_{i=1}^{n}\mathbb{1}_{T_i>t}},$$

where c is a threshold for the predicted output. The ROC curve for the full spectrum of sensitivities and specificities over $c\in[0,1]$, defined as

$$\mathrm{ROC}(t_\mathrm{P},\,t) = \{\mathrm{TPR}_{[t_\mathrm{P},\,t]}(c),\,1-\mathrm{TNR}_{[t_\mathrm{P},\,t]}(c)\,;\,c\in\mathbb{R}\}.$$

Area under the receiver operating characteristic curve (AUC) (Heagerty et al., 2000; Heagerty and Zheng, 2005), which measures the area beneath the ROC curve and provides a single scalar value to represent model performance, is then defined as

$$\mathrm{AUC}(t_\mathrm{P},\,t) = \int_0^1 \mathrm{TPR}_{[t_\mathrm{P},\,t]}\big((1-\mathrm{TNR}_{[t_\mathrm{P},\,t]}(p))^{-1}\big)\,dp.$$

Calibration

Brier Score (Brier, 1950) is a common metric to evaluate the similarity between the predicted and observed probabilities (derived from the observed time) on the same individual. In a dynamic prediction context, if an individual i whose survival time is in the time interval $[t_\mathrm{P},\,t]$ then the survival prediction $\pi_i(t\,|\,t_\mathrm{P})$ should be close to 1, otherwise if this individual has the event occur after t, the survival prediction should be close to 0. The Brier score

$BS(t_{\mathrm{P}}, t)$ is then defined as

$$BS(t_{\mathrm{P}}, t) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{T_i \in [t_{\mathrm{P}}, t], \Delta_i = 1} \pi_i(t \mid t_{\mathrm{P}})^2 + \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{T_i > t} \big(1 - \pi_i(t \mid t_{\mathrm{P}})\big)^2.$$

Integrated Brier Score provides an overall calculation of the Brier score at all available prediction times in the interval $[t_1, t_2]$, which writes

$$\mathrm{IBS} = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \mathrm{BS}(u, u + \delta_t) du.$$

2.5 Contributions

The rising availability of high-frequency longitudinal data in modern datasets presents valuable opportunities to improve survival models by enabling dynamic, real-time updates to risk assessments. However, this also brings several challenges. To address the challenges of integrating longitudinal data into survival models, especially in high-dimensional contexts, our contributions propose solutions that leverage advanced mathematical models, including joint modeling frameworks, feature extraction techniques, and regularization strategies. These approaches ensure the efficient processing and incorporation of this data into survival models, leading to improved predictive performance and more meaningful risk assessments.

We outline below the key contributions of this thesis. In the following, we consider a population of n individuals, with the data for each individual represented as follows:

$$\mathcal{D}_n = \{(T_1, \Delta_1, W_1, X_1), \ldots, (T_n, \Delta_n, W_n, X_n)\},$$

where T_i denotes the observed time, Δ_i denotes the event indicator, W_i denotes the static features and $X_i = \big(X_i(t_i^1), \ldots, X_i(t_i^{n_i})\big) \in \mathbb{R}^{d \times n_i}$ denotes the entire history of the d observed longitudinal markers at n_i time points $t_i^1 \leq \ldots \leq t_i^{n_i} \leq T_i$ up to its observed time T_i.

FLASH - joint models. We introduce in Chapter 3 a new joint model called FLASH (Fast joint model for Longitudinal And Survival data in High dimension), together with an efficient inference methodology. The model is inspired by both JLCMs and SREMs, but is specifically designed to handle high-dimensional longitudinal markers. In this model, the population $\mathcal{D}_n$ is assumed to consist of $K \in \mathbb{N}^*$ latent classes, each representing a different risk level. To each individual $i \in \{1, \ldots, n\}$, we associate a categorical latent variable $g_i \in \{1, \ldots, K\}$, which encodes its latent class membership. Then, the latent class

membership probability is assumed to take the form, for any $k \in \{1, \dots, K\}$,

$$\mathbb{P}(g_i = k) = \frac{e^{{W_i}^\top \xi_k}}{\sum_{j=1}^{K} e^{{W_i}^\top \xi_j}},$$

where $\xi_k \in \mathbb{R}^p$ denotes a vector of coefficients for class k.

The class-specific longitudinal model is described by a standard linear mixed model (Laird and Ware, 1982)

$$X_i(t)|g_i = k \sim \mathcal{N}\big(U(t)\beta_k + V(t)b_i, \Sigma\big),$$

where Σ is a variance-covariance matrix of measurement errors, $U(t) \in \mathbb{R}^{d\times q}$ is a matrix of time-varying features with corresponding unknown fixed effect parameters $\beta_k \in \mathbb{R}^q$, and $V(t) \in \mathbb{R}^{d\times r}$ is a matrix of time-varying features with corresponding random effect $b_i \sim \mathcal{N}(0, D)$, with $D \in \mathbb{R}^{r\times r}$ being a the variance-covariance matrix of random effects.

We denote by $X_{i,[0,t]} = \Big(X_i\big(t_i^1\big), \dots, X_i\big(t_i^{n_i(t)}\big)\Big)$ the longitudinal measurements recorded up to time t and $t_i^{n_i(t)}$ is the last recording time for longitudinal features before t. To quantify the effect of this longitudinal data on the survival time, it is represented by a set of $M \in \mathbb{N}^+$ fixed functionals $\Psi_m : X_{i,[0,t]} \to \Psi_m(X_{i,[0,t]}) \in \mathbb{R}^{d'}$, $m \in \{1, \dots, M\}$, where d' is the dimension of the extracted features and will vary depending on the type of functional. The class-specific survival model, which takes the form of a Cox model (Cox, 1972b), is then defined as

$$\lambda(t \mid X_{i,[0,t]}, g_i = k) = \lambda_0(t) \exp\Big(\sum_{m=1}^{M} \Psi_m\big(X_{i,[0,t]}\big)\gamma_{k,m}\Big), \tag{2.17}$$

where λ_0 is an unspecified baseline hazard function that does not depend on k and $\gamma_k = (\gamma_{k,1}, \dots, \gamma_{k,M})$ are the joint representation parameters, which are the only class-specific objects in this model.

A key distinction from SREMs is that the association features $\big(\Psi_1(X_{i,[0,t]}), \dots, \Psi_M(X_{i,[0,t]})\big)$ are assumed to be progressively independent of the modeling assumptions in the longitudinal submodel. This assumption of independence allows the model to be trained very efficiently, even with high-dimensional longitudinal data, since the likelihood is then in closed-form in Gaussian settings and therefore does not require Monte Carlo approximations.

Moreover, it allows the use of high-dimensional and generic feature extraction functions that characterize longitudinal markers, rather than just random effects, resulting in a model that is generic enough to be adapted to different types of longitudinal markers and prior information on the problem. The use of the elastic net and sparse group lasso regularization enables the automatic selection of relevant association features, resultin g in an interpretable model that retains only the significant longitudinal markers. In addition, our model automatically identifies significant features that are relevant from a practical point of view, making it interpretable, which is of the greatest importance for a prognostic algorithm in healthcare.

Finally, the model allows us to define a “real-time” prediction methodology, enabling

risk estimation at a given point in time for each subject using time-independent and longitudinal data available up to that time point. Real-time numerical experiments on both simulated and real data are given to compare the proposed framework with the main competing state-of-the-art methods for the subject. Our proposed framework has both better predictive performance and less computational complexity.

Modelling Point Processes with Controlled Latent States. In Chapter 4, we present a new framework based on neural networks for learning individual-specific intensities of counting processes from a set of static variables and longitudinal data. In this framework, the intensity function is parameterized as

$$\lambda_{i;\theta}(t) = \exp(\alpha^\top z_{i;\theta}(t) + \beta^\top W_i), \tag{2.18}$$

where $z_{i;\theta}(t) \in \mathbb{R}^p$ is a learned embedding of the time series $X_{i,[0,t]}$ parameterized by θ and (α, β) are learnable parameters.

The first major innovation is to leverage the framework of neural differential equations (Chen et al., 2018) to construct the embedding z_θ so that it captures complex dynamics of longitudinal data. In particular, we use the neural controlled differential equation (Kidger et al., 2020), which is a generalization of neural ODE. Then, the embedding $z_{i;\theta}(t)$ evolves according to the following differential equation:

$$dz_{i;\theta}(t) = \mathbf{G}_\psi\big(z_{i;\theta}(t)\big)dX_i(t)$$

with initial condition $z_{i;\theta}(0) = \mathbf{0}$. The function $\mathbf{G}_\psi : \mathbb{R}^p \to \mathbb{R}^{p\times d}$ is typically modeled as a small feed-forward neural network parameterized by ψ, and the overall learnable parameters are $\theta = (\alpha, \psi, \beta)$. Note that, the neural controlled differential equation allows for the latent state $z_{i;\theta}(t)$ to depend explicitly on the longitudinal feature $X_i(t)$ thus encoding richer dynamics.

Second, we explore the use of signatures (introduced in Section 2.2.4) to define z_θ. This approach builds on the connection between signatures and controlled differential equations, where a linear approximation of the solution to a CDE can be expressed as a linear function of the signatures (Bleistein et al., 2023). Consequently, the embedding $z_{i;\theta}(t)$ can be defined as

$$z_{i;\theta}(t) = \gamma^\top \mathbf{S}_N(X_i^{\text{sig}}(t)),$$

where the overall learnable parameters are $\theta = (\alpha, \beta, \gamma)$, $N \geq 1$ is treated as a hyperparameter, and $X_i^{\text{sig}}(t) = \big\{\big(t_i^1, X_i(t_i^1)\big), \ldots, \big(t, X_i(t)\big)\big\}$ is the extension of $X_{i,[0,t]}$ by adding the time dimension.

We demonstrate the strong performance of our models on a vast array of simulated and real-world survival analysis datasets from finance, predictive maintenance, and food supply chain management.

Comparison of classification and survival models for dynamic churn prediction. Chapter 5 presents the development of churn prediction models using dynamic client data

in non-contractual settings, specifically applied to Califrais' data. To build efficient models for this problem, we propose a comprehensive method by training models across three approaches.

First, we frame churn prediction as a binary classification problem, aiming to directly identify whether a customer is likely to churn within a certain period. This method is simple and easily understandable for non-experts, making it an effective choice for businesses looking to quickly assess churn risk. Second, we explore various survival analysis frameworks, including the two advanced methods introduced in Chapters 3 and 4, to not only predict whether a customer will churn but also estimate the time until they churn. This approach also provides more meaningful insights, enabling companies to implement targeted interventions before customers churn. Third, we extend the application of survival analysis by considering a landmark setting, which simplifies the training process and is better suited for high-dimensional datasets.

For each of these approaches, we implement a range of models and advanced feature engineering techniques to optimize predictive performance. Finally, we conduct a comprehensive comparison of these methodologies, evaluating their strengths and weaknesses in the context of real-time churn prediction. This comparative analysis across the three approaches, which is novel in the existing literature, provides valuable insights into the efficacy of different approaches, offering guidance for both academic research and practical application in churn management.

Outline of the manuscript The rest of the manuscript is organized as follows.

- Chapter 3 is a joint work with Adeline Fermanian (Califrais), Antoine Barbieri (Université Bordeaux), Sarah Zohar (INRIA), Anne-Sophie Jannot (INRIA), Simon Bussy (Califrais), and Agathe Guilloux (INRIA). It has been accepted for publication in *Biometrics*.
- Chapter 4 is a joint work with Linus Bleistein (INRIA), Adeline Fermanian (Califrais), and Agathe Guilloux (INRIA). In this work, my main contribution is implementation and running experiments. It has been published in *2024 International Conference on Machine Learning*.
- Chapter 5 is an ongoing joint work with Adeline Fermanian (Califrais), and Agathe Guilloux (INRIA).

Résumé détaillé

La disponibilité croissante des données longitudinales dans les données modernes offre des opportunités précieuses pour améliorer les modèles de survie, en permettant des mises à jour dynamiques et en temps réel des évaluations de risque. Cependant, cela entraîne également plusieurs défis. Nous nous concentrons sur ceux liés à l'intégration des données longitudinales dans les modèles de survie, en particulier dans des contextes à haute dimensionnalité. Nos contributions proposent des solutions qui tirent parti de modèles mathématiques avancés, y compris des cadres de modélisation conjointe, des techniques

d'extraction de caractéristiques et des stratégies de régularisation. Ces approches garantissent un traitement efficace et une incorporation de ces données dans les modèles de survie, ce qui conduit à une amélioration des performances prédictives et à des évaluations de risque plus significatives.

Nous exposons ci-dessous les principales contributions de cette thèse. Nous considérons une population de n individus, avec les données de chaque individu représentées comme suit :

$$\mathcal{D}_n = \{(T_1, \Delta_1, W_1, X_1), \ldots, (T_n, \Delta_n, W_n, X_n)\},$$

où T_i désigne le temps observé, Δ_i l'indicateur d'événement, W_i les caractéristiques statiques et $X_i = \big(X_i(t_i^1), \ldots, X_i(t_i^{n_i})\big) \in \mathbb{R}^{d \times n_i}$ représente l'intégralité de l'historique des d marqueurs longitudinaux observés à n_i temps $t_i^1 \leq \ldots \leq t_i^{n_i} \leq T_i$ jusqu'à son temps observé T_i.

FLASH - modèles conjoints Nous introduisons dans le chapitre 3 un nouveau modèle conjoint appelé FLASH (Fast joint model for Longitudinal And Survival data in High dimension), accompagné d'une méthodologie d'inférence efficace. Le modèle s'inspire des JLCMs et des SREMs, mais est spécifiquement conçu pour traiter des marqueurs longitudinaux de haute dimension. Dans ce modèle, la population $\mathcal{D}_n$ est supposée être constituée de $K \in \mathbb{N}^*$ classes latentes, chacune représentant un niveau de risque différent. À chaque individu $i \in \{1, \ldots, n\}$, nous associons une variable latente catégorique $g_i \in \{1, \ldots, K\}$, qui encode son appartenance à une classe latente. Ensuite, la probabilité d'appartenance à une classe latente prend la forme suivante, pour tout $k \in \{1, \ldots, K\}$:

$$\mathbb{P}(g_i = k) = \frac{e^{{W_i}^\top \xi_k}}{\sum_{j=1}^{K} e^{{W_i}^\top \xi_j}},$$

où $\xi_k \in \mathbb{R}^p$ désigne un vecteur de coefficients pour la classe k.

Dans chaque classe, le modèle longitudinal spécifique est décrit par un modèle linéaire à effets mixtes standard (Laird and Ware, 1982) :

$$X_i(t)|g_i = k \sim \mathcal{N}\big(U(t)\beta_k + V(t)b_i, \Sigma\big),$$

où Σ est une matrice de variance-covariance des erreurs de mesure, $U(t) \in \mathbb{R}^{d \times q}$ est une matrice de caractéristiques variant dans le temps avec des paramètres d'effets fixes inconnus $\beta_k \in \mathbb{R}^q$, et $V(t) \in \mathbb{R}^{d \times r}$ est une matrice de caractéristiques variant dans le temps avec un effet aléatoire correspondant $b_i \sim \mathcal{N}(0, D)$, où $D \in \mathbb{R}^{r \times r}$ est la matrice de variance-covariance des effets aléatoires.

Nous notons par $X_{i,[0,t]} = \Big(X_i\big(t_i^1\big), \ldots, X_i\big(t_i^{n_i(t)}\big)\Big)$ les mesures longitudinales enregistrées jusqu'au temps t, et $t_i^{n_i(t)}$ est le dernier instant d'enregistrement des caractéristiques longitudinales avant t. Pour quantifier l'effet de ces données longitudinales sur le temps de survie, elles sont représentées par un ensemble de $M \in \mathbb{N}^+$ fonctionnelles fixes $\Psi_m : X_{i,[0,t]} \to \Psi_m(X_{i,[0,t]}) \in \mathbb{R}^{d'}$, $m \in \{1, \ldots, M\}$, où d' est la dimension des caractéristiques extraites et varie en fonction du type de fonctionnelle. Le modèle de survie spécifique

à chaque classe, qui prend la forme d'un modèle de Cox (Cox, 1972b), est alors défini par :

$$\lambda(t \mid X_{i,[0,t]}, g_i = k) = \lambda_0(t) \exp\Big(\sum_{m=1}^{M} \Psi_m\big(X_{i,[0,t]}\big)\gamma_{k,m} \Big),$$

où λ_0 est une fonction de risque de base non spécifiée qui ne dépend pas de k et $\gamma_k = (\gamma_{k,1}, \ldots, \gamma_{k,M})$ sont les paramètres de représentation conjointe, qui sont les seuls objets spécifiques à la classe dans ce modèle.

Une distinction clé par rapport aux SREMs est que les caractéristiques d'association $\big(\Psi_1(X_{i,[0,t]}), \ldots, \Psi_M(X_{i,[0,t]})\big)$ sont supposées être indépendantes de manière progressive des hypothèses de modélisation dans le sous-modèle longitudinal. Cette hypothèse d'indépendance permet d'entraîner le modèle de manière très efficace, même avec des données longitudinales de haute dimension, car la vraisemblance est alors explicites dans des contextes gaussiens et ne nécessite donc pas d'approximations de Monte Carlo.

De plus, elle permet d'utiliser des fonctions d'extraction de caractéristiques génériques et de haute dimension qui caractérisent les marqueurs longitudinaux, plutôt que de se limiter aux effets aléatoires, ce qui rend le modèle suffisamment générique pour être adapté à différents types de marqueurs longitudinaux et à l'information préalable sur le problème. L'utilisation de la régularisation Elastic Net et du group-lasso permet la sélection automatique des caractéristiques d'association pertinentes, aboutissant à un modèle interprétable qui conserve uniquement les marqueurs longitudinaux significatifs. De plus, notre modèle identifie automatiquement les caractéristiques significatives sur le plan pratique, ce qui le rend interprétable, ce qui est d'une grande importance pour un algorithme pronostique en santé.

Enfin, le modèle permet de définir une méthodologie de prédiction "en temps réel", permettant une estimation des risques à un moment donné pour chaque sujet, en utilisant les données longitudinales et indépendantes du temps disponibles jusqu'à ce point temporel. Des expériences numériques en temps réel sur des données simulées et réelles sont présentées pour comparer le cadre proposé avec les principales méthodes concurrentes de l'état de l'art pour le sujet. Le cadre proposé présente à la fois de meilleures performances prédictives et une moindre complexité computationnelle.

Modélisation des processus de comptage avec des états latents contrôlés Dans le chapitre 4, nous présentons un nouveau cadre basé sur les réseaux neuronaux pour apprendre les intensités spécifiques aux individus de processus de comptage à partir d'un ensemble de variables statiques et de données longitudinales. Dans ce cadre, la fonction d'intensité est paramétrée par :

$$\lambda_{i;\theta}(t) = \exp(\alpha^\top z_{i;\theta}(t) + \beta^\top W_i),$$

où $z_{i;\theta}(t) \in \mathbb{R}^p$ est un encodage appris de la série temporelle $X_{i,[0,t]}$ paramétré par θ, et (α, β).

La première grande innovation est d'exploiter le cadre des équations différentielles neuronales (Chen et al., 2018) pour construire l'encodage z_θ afin qu'il capture les dynamiques

complexes des données longitudinales. En particulier, nous utilisons les équations différentielles contrôlées neuronales (CDE), une généralisation des équations différentielles ordinaires neuronales (ODE), comme le montre le travail de Kidger et al. (2020). L'encodage $z_{i;\theta}(t)$ évolue selon l'équation différentielle suivante :

$$dz_{i;\theta}(t) = \mathbf{G}_\psi\big(z_{i;\theta}(t)\big)dX$$

où $\mathbf{G}_\psi$ est une fonction qui paramètre l'évolution des représentations des données longitudinales sur le temps, et $dX_i(t)$ est l'incrément des observations longitudinales à chaque point t. Cette formulation permet d'obtenir une représentation dynamique et continue de l'historique longitudinal, capturant ainsi les complexités temporelles et inter-individuelles des données de manière plus flexible que les approches statiques classiques.

Deuxièmement, nous explorons l'utilisation des signatures (introduites dans la section 2.2.4) pour définir z_θ. Cette approche repose sur la connexion entre les signatures et les équations différentielles contrôlées, où une approximation linéaire de la solution d'une CDE peut être exprimée comme une fonction linéaire des signatures (Bleistein et al., 2023). Par conséquent, l'encodage $z_{i;\theta}(t)$ peut être défini comme suit :

$$z_{i;\theta}(t) = \gamma^\top \mathbf{S}_N(X_i^{\text{sig}}(t)),$$

où les paramètres à apprendre sont $\theta = (\alpha, \beta, \gamma)$, $N \geq 1$ est traité comme un hyperparamètre, et $X_i^{\text{sig}}(t) = \big\{\big(t_i^1, X_i(t_i^1)\big), \ldots, \big(t, X_i(t)\big)\big\}$ est l'extension de $X_{i,[0,t]}$ en ajoutant la dimension temporelle.

Nous démontrons les excellentes performances de nos modèles sur un large éventail de jeux de données de survie simulées et réelles, provenant de secteurs tels que la finance, la maintenance prédictive et la gestion des chaînes d'approvisionnement alimentaires.

Comparaison des modèles de classification et de survie pour la prédiction dynamique du churn Le chapitre 5 présente le développement de modèles de prédiction du churn utilisant des données historiques de clients, spécifiquement appliqués aux données de Califrais. Pour construire des modèles efficaces pour ce problème, nous entraînons des modèles selon trois approches.

Premièrement, nous abordons la prédiction du churn comme un problème de classification binaire, visant à identifier directement si un client est susceptible de se désabonner dans un certain délai. Cette méthode est simple et facilement compréhensible par des non-experts, ce qui en fait un choix efficace pour les entreprises cherchant à évaluer rapidement le risque de churn. Deuxièmement, nous explorons différents cadres d'analyse de survie, y compris les deux méthodes avancées introduites dans les chapitres 3 et 4, afin de prédire non seulement si un client se désabonnera, mais aussi d'estimer le temps avant qu'il ne se désabonne. Cette approche permet également d'obtenir des informations plus significatives, offrant aux entreprises la possibilité de mettre en œuvre des interventions ciblées avant que les clients ne se désabonnent. Troisièmement, nous étendons l'application de l'analyse de survie en considérant un cadre de repère, ce qui simplifie le processus d'entraînement et est mieux adapté aux ensembles de données de haute dimension.

Pour chacune de ces approches, nous mettons en œuvre une gamme de modèles et de

techniques avancées d'ingénierie des caractéristiques pour optimiser les performances prédictives. Enfin, nous menons une comparaison approfondie de ces méthodologies, en évaluant leurs forces et leurs faiblesses dans le contexte de la prédiction du churn en temps réel. Cette analyse comparative entre les trois approches fournit des informations précieuses sur l'efficacité de chaque approche, offrant des orientations tant pour la recherche académique que pour l'application pratique dans la gestion du churn.

- Le chapitre 3 est un travail joint avec Adeline Fermanian (Califrais), Antoine Barbieri (Université Bordeaux), Sarah Zohar (INRIA), Anne-Sophie Jannot (INRIA), Simon Bussy (Califrais), and Agathe Guilloux (INRIA). Il est accepté pour publication dans *Biometrics*.
- Le chapitre 4 est un travail joint avec Linus Bleistein (INRIA), Adeline Fermanian (Califrais), and Agathe Guilloux (INRIA). Il a été publié dans *2024 International Conference on Machine Learning*.

Chapter 3

An efficient joint model for high dimensional longitudinal and survival data via generic association features

Contents

3.1 Introduction

In healthcare, it is increasingly common to record the values of longitudinal features (e.g. biomarkers such as heart rate or hemoglobin level) up to the occurrence of an event of interest for a subject, such as rehospitalization, relapse, or disease progression. Moreover, in large observational databases such as claims databases, with electronic health records, the amount of recorded data per patient is often very large and growing over time.

However, there is currently no tool that can simultaneously process high-dimensional longitudinal signals and perform real-time predictions, i.e. give predictions at any time after only one estimation/training step. While landmark approaches (see, e.g., Devaux et al., 2022) can handle high-dimensional longitudinal features, they require separate training for each prediction time. An alternative is to use "joint modeling" to handle longitudinal and survival outcomes together.

The latter has received considerable attention in the last two decades (Tsiatis and Davidian, 2004; Rizopoulos and Ghosh, 2011; Hickey et al., 2016). More specifically, it consists of defining (i) a time-to-event model, (ii) a longitudinal marker model, and (iii) linking both models via a common latent structure. Numerical studies suggest that these approaches are among the most satisfactory for incorporating all longitudinal information into a survival model (Yu et al., 2004), and are better than landmark approaches, which use only information from individuals at risk at the landmark time (see, e.g., Devaux et al., 2022). They have the additional advantage of making more efficient use of the data as information on survival is also used to model the longitudinal markers. More importantly, they require only one training, regardless of the number of prediction times.

There are two main approaches to linking longitudinal and survival models. On the one hand, in shared parameter joint models (SREMs), characteristics of the longitudinal marker, typically some random effects learned in a linear mixed model, are included as covariates in the survival model (Wulfsohn and Tsiatis, 1997; Andrinopoulou and Rizopoulos, 2016). On the other hand, joint latent class models (JLCMs), inspired by mixture-of-experts modelling (Masoudnia and Ebrahimpour, 2014), assume that the dependence between the time-to-event and the longitudinal marker is fully captured by a latent class structure (Lin et al., 2002b; Proust-Lima et al., 2014), which amounts to assuming that the population is heterogeneous and that there are homogeneous latent classes that share the same marker trajectories and prognosis. JLCMs offer a computationally attractive alternative to SREMs, especially in a high-dimensional context. These two models are illustrated in Figure 3.1.

Unfortunately, joint models have predominantly focused on univariate longitudinal markers (Andrinopoulou et al., 2020). To adapt such models to a multivariate setting, the common approach is to fit multiple univariate joint models separately to each longitudinal marker (Wang et al., 2012), which does not account for interactions between longitudinal markers (Jaffa et al., 2014; Kang and Song, 2022; Lin et al., 2002a). Furthermore, issues arising from the high-dimensional context—e.g. computational power, limits of numerical estimation—have, to our knowledge, never been considered in the analyses, and the number of longitudinal markers considered in numerical studies remains very low, typically up to 5 (Hickey et al., 2016; Murray and Philipson, 2022; Rustand et al., 2024).

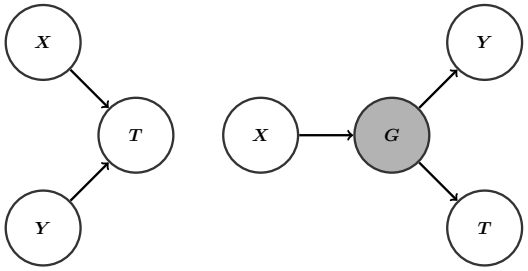

Figure 3.1 – Graphical representation of SREMs (left) and JLCMs (right). The variable X represents time-independent features, Y the longitudinal markers, T the time-to-event, and G the latent class membership.

The aim of this article is to propose a new joint model called FLASH (Fast joint model for Longitudinal And Survival data in High dimension), together with an efficient inference methodology. The model is inspired by both JLCMs and SREMs, but is designed to scale to high-dimensional longitudinal markers. The general idea is to use generic features extracted from the longitudinal markers directly in the survival model and to use regularization in an Expectation-Maximization (EM) algorithm. The main difference with SREMs is that these features, called association features, are assumed to be independent of the modeling assumptions in the longitudinal submodel. As a result, the model is very efficient to train—the likelihood is closed-form in a Gaussian setting and does not require Monte Carlo approximations, as is often the case with SREMs, making it suitable for high-dimensional longitudinal markers. Moreover, it allows the use of high-dimensional and generic feature extraction functions that characterize longitudinal markers, rather than just the random effects, resulting in a model that is generic enough to be adapted to different types of longitudinal markers and prior information on the problem. The use of elastic net and sparse group lasso regularization enables automatic selection of relevant association features, resulting in an interpretable model that retains only the significant longitudinal markers

Finally, our model allows us to define a "real-time" prediction methodology where, once the parameters of the model have been learned, we can compute a predictive marker that, given only the time-independent and longitudinal features up to a given point in time, outputs a risk for each subject at that point in time. It differs from traditional approaches (Proust-Lima et al., 2014) that require knowledge of the survival labels, which are unknown in the "real-time" prediction setting.

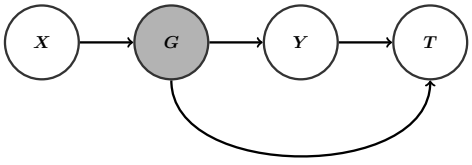

Figure 3.2 – Graphical representation of the FLASH model. The variable X represents time-independent features, Y the longitudinal markers, T the time-to-event, and G the latent class membership of an individual.

In summary, we introduce a new method for predicting survival risk with high-dimensional longitudinal features that is both interpretable and computationally efficient, thus providing

a powerful tool for real-time clinical decision making, for example in patient monitoring.

A precise description of the model is given in Section 3.2. In Section 3.3, we present our assumptions and inference methodology based on maximizing the likelihood with a regularized variant of the EM algorithm. Section 3.4 introduces our evaluation methodology and the competing methods. In Section 3.5, we apply our method to datasets from two simulation studies and three publicly available medical datasets (PBCseq, AIDS, Sepsis). We show that FLASH outperforms the competitors and selects relevant features from a medical perspective. Finally, we discuss the obtained results in Section 3.6. The code to implement the model and reproduce the experiments is publicly available at *https://github.com/Califrais/flash.git*.

3.2 Model

In this section, we describe the FLASH model, which consists of three sub-models: a multinomial logistic regression defining the probability of belonging to a latent class, a generalized linear mixed model for each latent class describing the evolution of the longitudinal markers, and finally a Cox class-specific survival model. In all the following, we consider a set of n independent and identically distributed (i.i.d.) subjects. For each subject $i \in \{1, \dots, n\}$ we are given some longitudinal markers Y_i, time-independent features $X_i \in \mathbb{R}^p$, a right-censored time-to-event $T_i \in \mathbb{R}^+$, and a censoring variable $\Delta_i \in \{0, 1\}$.

3.2.1 Latent class membership

We assume that the population consists of $K \in \mathbb{N}^*$ latent groups. To each subject $i \in \{1, \dots, n\}$, we associate a categorical latent variable $g_i \in \{1, \dots, K\}$, which encodes its latent class membership. Then, denoting by $X_i \in \mathbb{R}^p$ the p-dimensional vector of time-independent features, the latent class membership probability is assumed to take the multinomial logistic regression form (Böhning, 1992), for any $k \in \{1, \dots, K\}$,

$$\mathbb{P}(g_i = k) = \frac{e^{X_i^\top \xi_k}}{\sum_{j=1}^{K} e^{X_i^\top \xi_j}}, \tag{3.1}$$

where $\xi_k \in \mathbb{R}^p$ denotes a vector of coefficients for class k. This submodel is similar to JLCMs or the C-mix model (Bussy et al., 2019a) and assumes that latent-class membership depends only on time-independent features, with the vector ξ_k quantifying the effect of each time-independent feature in X_i on the probability that subject i belongs to the k-th latent class. The optimal number of latent classes K can be selected with the Bayesian information criterion (BIC) (Hastie et al., 2009), see Section A.3.8 of the Supplementary Materials for more details. Note that modeling the probability of latent class membership using multinomial logistic regression is a common assumption in joint models (Lin et al., 2002b; Proust-Lima et al., 2014). However, other models such as hidden Markov models can also be used to represent this probability (Bartolucci and Farcomeni, 2015; Bartolucci

and Farcomeni, 2019). We assess the sensitivity to this choice by some experiments in Section A.3.9 of the Supplementary Materials.

3.2.2 Class-specific longitudinal model

For each subject $i \in \{1, \dots, n\}$, we are given $L \in \mathbb{N}^*$ longitudinal markers. We let, for any $\ell \in \{1, \dots, L\}$, $Y_i^\ell = \left(y_i^\ell(t_{i1}^\ell), \dots, y_i^\ell(t_{in_i^\ell}^\ell)\right)^\top \in \mathbb{R}^{n_i^\ell}$ be the vector of repeated measures of a theoretical longitudinal marker y_i^ℓ at observation times (or follow-up visits) $0 \leq t_{i1}^\ell < \cdots < t_{in_i^\ell}^\ell$. Note that the observation times t_{ij}^ℓ, $j = 1, \dots, n_i^\ell$, can differ between subjects as well as between longitudinal markers, which makes the assumptions on the sampling mechanism very weak. In particular, this setting encapsulates many scenarios considered as missing data, where one individual is not measured while another one is, or where the longitudinal maker of one individual is missing, which here both simply correspond to removing some timestamps from the corresponding grids.

We assume a class-specific generalized linear mixed model (GLMM) for each longitudinal marker, which is a classical model for longitudinal data (Fitzmaurice et al., 2012; Hickey et al., 2016). The GLMM is chosen according to the nature of the markers: Gaussian linear model for continuous markers, logistic regression for a binary marker, and Poisson regression for counts. For the continuous markers, given a latent class $g_i = k$, for the ℓ-th marker at time $t \in \{t_{i1}^\ell, \dots, t_{in_i^\ell}^\ell\}$, we then have

$$y_i^\ell(t_{ij}^\ell) \,|\, b_i^\ell,\, g_i = k \sim \mathcal{N}(m_{ik}^\ell(t_{ij}^\ell), \phi_\ell), \tag{3.2}$$

where the variance $\phi_\ell \in \mathbb{R}^+$ is an estimated parameter and the mean m_{ik}^ℓ is defined by

$$m_{ik}^\ell(t) = u^\ell(t)^\top \beta_k^\ell + v^\ell(t)^\top b_i^\ell,$$

where $u^\ell(t) \in \mathbb{R}^{q_\ell}$ is a row vector of time-varying features with corresponding unknown fixed effect parameters $\beta_k^\ell \in \mathbb{R}^{q_\ell}$, and $v^\ell(t) \in \mathbb{R}^{r_\ell}$ is a row vector of time-varying features with corresponding random effect $b_i^\ell \in \mathbb{R}^{r_\ell}$. Flexible representations for $u^\ell(t)$ can be considered using a vector of time monomials $u^\ell(t) = (1, t, t^2, \dots, t^\alpha)^\top$, with $\alpha \in \mathbb{N}^+$. We use $\alpha = 1$ in all our experiments but higher orders could be used. We also let $v^\ell(t) = (1, t)^\top$.

Classically, the random effects component is assumed to follow a zero-mean multivariate normal distribution, that is, $b_i^\ell \sim \mathcal{N}(0, D^{\ell\ell})$ with $D^{\ell\ell} \in \mathbb{R}^{r_\ell \times r_\ell}$ the variance-covariance matrix. To account for the dependence between the different longitudinal markers, we let $\mathrm{Cov}[b_i^\ell, b_i^{\ell'}] = D^{\ell\ell'}$ for $\ell \neq \ell'$, where $\mathrm{Cov}[\cdot, \cdot]$ denotes the covariance matrix of two random vectors, and we denote by $D = (D^{\ell\ell'})_{1 \leq \ell, \ell' \leq L}$ the global variance-covariance matrix which is common to all latent classes. Note that this variance–covariance matrix D can be easily extended to be class-specific. We assume that all dependencies between longitudinal markers are encapsulated in this matrix D, which is summarized in the following assumption.

Assumption 1. For any $\ell \in \{1, \dots, L\}$ and any $i \in \{1, \dots, n\}$, the longitudinal markers Y_i^ℓ are pairwise independent conditionally on b_i^ℓ and g_i.

This is a standard modelling assumption in joint models (see, e.g., Tsiatis and Davidian, 2004). Then, if we concatenate all longitudinal measurements and random effects of subject i in, respectively, $Y_i = \left(Y_i^{1\top} \cdots Y_i^{L\top}\right)^\top \in \mathbb{R}^{n_i}$ and $b_i = \left(b_i^{1\top} \cdots b_i^{L\top}\right)^\top \in \mathbb{R}^r$ with $n_i = \sum_{\ell=1}^L n_i^\ell$ and $r = \sum_{\ell=1}^L r_\ell$, a consequence of Assumption 1 and Equation (3.2) is that

$$Y_i \,|\, b_i, g_i = k \sim \mathcal{N}\big(M_{ik}, \Sigma_i\big), \tag{3.3}$$

where $M_{ik} = \left(m_{ik}^1(t_{i1}^1), \ldots, m_{ik}^1(t_{in_i^1}^1), \ldots, m_{ik}^L(t_{i1}^L), \ldots, m_{ik}^L(t_{in_i^L}^L)\right)^\top \in \mathbb{R}^{n_i}$ and Σ_i is the diagonal matrix whose diagonal is $(\phi_1 \mathbf{1}_{n_i^1}{}^\top, \ldots, \phi_L \mathbf{1}_{n_i^L}{}^\top)^\top \in \mathbb{R}^{n_i}$ where $\mathbf{1}_m$ denotes the vector of $\mathbb{R}^m$ having all coordinates equal to one.

To extend the GLMM to other types of longitudinal markers (for example, binary or count data), the key point is that the distributions of b_i and of $Y_i \,|\, b_i, g_i = k$ should be conjugate, so that the likelihood is tractable. It is possible to extend the model to non-conjugate distributions via numerical integration methods but this is not immediate. A detailed discussion of this is given in Section A.1.1 of the Supplementary Materials. For simplicity, we restrict ourselves to the Gaussian case in this article.

3.2.3 Class-specific Cox survival model

We place ourselves in a classical survival analysis framework. Let the non-negative random variables $T_i^\star$ and C_i be the time to the event of interest and the censoring time, respectively. We then define the observed time $T_i = T_i^\star \wedge C_i$ and censoring indicator $\Delta_i = \mathbb{1}_{\{T_i^\star \leq C_i\}}$, where $a \wedge b$ denotes the minimum between two real numbers a and b, and $\mathbb{1}_{\{.\}}$ is the indicator function which takes the value 1 if the condition in $\{\cdot\}$ is satisfied, and 0 otherwise. We denote by $\mathcal{Y}_i^\ell(t^-) = \left(y_i^\ell(t_{i1}^\ell), \ldots, y_i^\ell(t_{iu}^\ell)\right)_{0 \leq t_{iu}^\ell < t}$ the subset of Y_i^ℓ formed from observations up to time t and by $\mathcal{Y}_i(t^-)$ the concatenation of the history of all observed longitudinal markers up to t. Then we consider $M \in \mathbb{N}^+$ user-defined feature extraction functions $\Psi_m : \mathcal{Y}_i^\ell(t^-) \to \Psi_m(\mathcal{Y}_i^\ell(t^-)) \in \mathbb{R}$, $m \in \{1, \ldots, M\}$, which characterise the longitudinal markers. The set of features $\left(\Psi_m(\mathcal{Y}_i^\ell(t^-))\right)_{1 \leq m \leq M}$ should be rich enough to capture all dependencies between longitudinal markers and time-to-event, and is discussed in more detail below. To quantify the effect of the longitudinal markers on time-to-event, we then use an extension of the Cox relative risk model (Cox, 1972a), which allows time-varying covariates and was firstly introduced in Andersen and Gill (1982). Note that this model does not fulfill the classical Cox model's assumption of a constant proportional hazard over time. The hazard function in this model takes the form

$$\lambda(t \,|\, \mathcal{Y}_i(t^-), g_i = k) = \lambda_0(t) \exp\Big(\sum_{\ell=1}^L \sum_{m=1}^M \Psi_m\big(\mathcal{Y}_i^\ell(t^-)\big)\gamma_{k,m}^\ell\Big) = \lambda_0(t) \exp\left(\psi_i(t)^\top \gamma_k\right), \tag{3.4}$$

where λ_0 is an unspecified baseline hazard function that does not depend on k, $\gamma_{k,m}^\ell \in \mathbb{R}$ the joint association parameters, which are the only class-specific objects in this model. We concatenate them in $\gamma_k = (\gamma_{k,1}^1, \ldots, \gamma_{k,M}^1, \ldots, \gamma_{k,1}^L, \ldots, \gamma_{k,M}^L)^\top \in \mathbb{R}^{LM}$ and define $\psi_i(t) = \Big(\Psi_1\big(\mathcal{Y}_i^1(t^-)\big), \ldots, \Psi_M\big(\mathcal{Y}_i^1(t^-)\big), \ldots, \Psi_1\big(\mathcal{Y}_i^L(t^-)\big), \ldots, \Psi_M\big(\mathcal{Y}_i^L(t^-)\big)\Big)^\top \in \mathbb{R}^{LM}$.

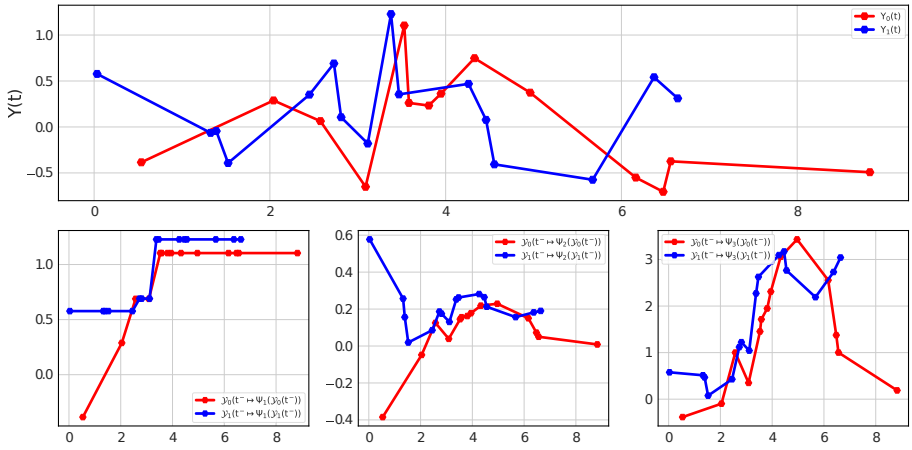

Figure 3.3 – On the **top**, observed longitudinal markers of 2 individuals from the FLASH_simu dataset. On the **bottom**, association features applied on the observed longitudinal markers: the maximum (**left**), the mean (**center**), and the sum (**right**).

This model can be viewed as a generalization of SREMs (Lin et al., 2002a; Rizopoulos and Ghosh, 2011), which have hazard functions of the form $\lambda_0(t)\exp\left(\sum_{\ell=1}^{L}\phi(b_i^\ell,t)^\top\gamma^\ell\right)$, where the association between the longitudinal and survival models is captured by the random effects b_i^ℓ. The key idea of our model lies in the following assumption.

Assumption 2. For any time $t \geq 0$, the hazard rate at time t conditionally on the history of the longitudinal markers up to t^- and g_i is independent of b_i.

Any feature extraction function Ψ_m of the longitudinal markers can be considered in the hazard function. Simple examples of such functions are the maximum or the sum (or the mean) of the longitudinal features, respectively defined by

$$\Psi_m(\mathcal{Y}_i^\ell(t^-)) = \max_{j:t_{ij}^\ell<t}\left\{y_i^\ell(t_{i1}^\ell),\ldots,y_i^\ell(t_{ij}^\ell)\right\}, \quad \Psi_m(\mathcal{Y}_i^\ell(t^-)) = \sum_{j:t_{ij}^\ell<t} y_i^\ell(t_{ij}^\ell).$$

For illustration, Figure 3.3 shows how these association features change with time. In practice, our rationale is to use a variety of feature extraction functions Ψ_m, such as absolute energy over time, statistics on autocorrelation, or Fourier and wavelet basis projections, and then perform feature selection via regularisation to learn which ones are predictive for the underlying task. This will be described in more detail in Section 3.3.2. Note that a crucial aspect of this model is that the extracted vector, also called extracted features or association features, $\psi_i(t)$, does not depend on the modelling assumptions in the longitudinal submodel of Subsection 3.2.2 — that is, does not depend on b_i other than through the history $\mathcal{Y}_i(t^-)$.

The FLASH model is summarised in Figure 3.2 which shows that our model is a combination of SREMs and JLCMs where both random effects and latent classes account for the dependence between longitudinal markers and time-to-event.

3.3 Inference

Now that we have introduced all the components of our model, in this section we derive the form of its likelihood, present the regularisation strategy that deals with the high dimensionality of the data, and finally present our variant of the EM algorithm used to minimise the penalised negative log-likelihood.

3.3.1 Likelihood

Consider a training cohort of n i.i.d. subjects $\mathcal{D}_n = \big((X_1, Y_1, T_1, \Delta_1), \dots, (X_n, Y_n, T_n, \Delta_n)\big)$. For simplicity, we slightly abuse notation and use the same notation $f^\star$ for the true (joint) density or probability mass function of the various random variables in our model. Similarly, we denote by f_θ the candidates for estimating the densities $f^\star$ that satisfy the model assumptions of Section 3.2, where we have concatenated in θ all $P \in \mathbb{N}^+$ unknown parameters:

$$\theta = \big(\xi_1^\top, \dots, \xi_K^\top, \beta_1^\top, \dots, \beta_K^\top, \phi^\top, D, \lambda_0(\tau_1), \dots, \lambda_0(\tau_J), \gamma_1^\top, \dots \gamma_K^\top\big)^\top \in \mathbb{R}^P,$$

where $\beta_k = (\beta_k^{1\top} \dots \beta_k^{\ell\top})^\top \in \mathbb{R}^q$ with $q = \sum_{\ell=1}^L q_\ell$ for any $k \in \{1, \dots, K\}$, $\phi = (\phi_1, \dots, \phi_L)^\top$ and where we use the vectorization of the matrix D although this is not written explicitly. Note that we classically (see, e.g., Klein, 1992) estimate λ_0 by a function taking mass at each failure time $\tau_j \in (\tau_1, \dots, \tau_J)$, where $(\tau_1, \dots, \tau_J)$ denote the $J \in \mathbb{N}^+$ unique failure times (obtained from $(T_1, \dots, T_n)$ removing the duplicates and keeping only the uncensored times T_i for which $\Delta_i = 1$). In this way, the estimation of the function λ_0 amounts to the estimation of the vector $\big(\lambda_0(\tau_1), \dots, \lambda_0(\tau_J)\big)$.

First, conditioning on the latent classes, we have

$$f^\star(T_i, \Delta_i, Y_i) = \sum_{k=1}^K f^\star(g_i = k) f^\star(T_i, \Delta_i | Y_i, g_i = k) f^\star(Y_i | g_i = k).$$

This yields the negative log-likelihood

$$\mathcal{L}_n(\theta) = -n^{-1} \sum_{i=1}^n \log \sum_{k=1}^K f_\theta(g_i = k) f_\theta(T_i, \Delta_i \,|\, Y_i, g_i = k) f_\theta(Y_i \,|\, g_i = k). \tag{3.5}$$

Assuming that both the censoring mechanism and the stochastic mechanism generating the observation times of the longitudinal markers are non-informative (Rizopoulos and Ghosh, 2011), the joint density of (T_i, Δ_i) can be factorized into a part depending on the distribution of $T_i^\star$ and a part depending on that of C_i, so that

$$\begin{aligned} f^\star(T_i, \Delta_i | Y_i, g_i = k) &\propto f^\star(T_i | Y_i, g_i = k)^{\Delta_i} S^\star(T_i | Y_i, g_i = k)^{1-\Delta_i} \\ &= \lambda^\star(T_i | Y_i, g_i = k)^{\Delta_i} S^\star(T_i | Y_i, g_i = k), \end{aligned} \tag{3.6}$$

where $S^\star$ and $\lambda^\star$ are the survival and hazard function associated with the density $f^\star$ of $T_i^\star$.

Under the assumptions given in the previous subsections, all terms in (3.5) can be computed in closed form. Indeed, $f_\theta(g_i = k)$ is given by (3.1) and the density function $f_\theta(Y_i|g_i = k)$ can be derived from the distribution of b_i and (3.3) (detailed calculations are given in Section A.1 of the Supplementary Materials). Furthermore, following Equation (3.6), we have $f_\theta(T_i, \Delta_i|Y_i, g_i = k) \propto \lambda\big(T_i \,|\, \mathcal{Y}_i(T_i^-), g_i = k\big)^{\Delta_i} S_k(T_i)$, where $S_k(t) = \exp\Big(-\int_0^t \lambda\big(s \,|\, \mathcal{Y}_i(s^-), g_i = k\big)\mathrm{d}s\Big)$ is the survival function of subject i given that it belongs to latent class k. Since the baseline hazard function λ_0 takes mass only at each failure time $\tau_j \in (\tau_1, \ldots, \tau_J)$ then the integration over the survival process $S_k(t)$ is simply a finite sum over the process evaluated at the J failure times. Then, we rewrite the function S_k, for any $t \geq 0$, as

$$S_k(t) = \exp\Big(-\sum_{j=1}^{J} \lambda(\tau_j)\mathbb{1}_{\{\tau_j \leq t\}}\Big) = \exp\Big(-\sum_{j=1}^{J} \lambda_0(\tau_j)\exp\big(\psi_i(\tau_j)^\top \gamma_k\big)\mathbb{1}_{\{\tau_j \leq t\}}\Big).$$

The fact that $f_\theta(T_i, \Delta_i|Y_i, g_i = k)$ is closed-form is one of the major advantages of our model over standard SREMs. Indeed, computing this density in SREMs usually requires integrating it with respect to the distribution of the random effects b_i, leading to intractable integrals in the log-likelihood function. These integrals are typically estimated using Monte Carlo techniques (Hickey et al., 2018), which are computationally intensive and require additional assumptions on the allowed association functions ψ_i. These approaches usually do not scale in a high-dimensional context.

To minimize (3.5) with respect to θ, we use the EM algorithm, which is the common choice in the literature (Wulfsohn and Tsiatis, 1997; Lin et al., 2002a). This requires deriving what we call the negative "complete" log-likelihood, that is, an estimation of the joint density $f^*(T_i, \Delta_i, Y_i, b_i, g_i)$, where the random effect b_i and the latent class g_i are not observed. To this end, we need the following independence assumption.

Assumption 3. For any $i \in \{1, \ldots, n\}$ and $\ell \in \{1, \ldots, L\}$, the random effects b_i^ℓ are independent of the latent class membership g_i, and remain independent of it conditionally on T_i, Δ_i, and Y_i.

This assumption states that subject-and-longitudinal marker specific random effects b_i^ℓ do not depend on the latent class membership. Then, we have

$$\begin{aligned} f^\star(T_i, \Delta_i, Y_i, b_i, g_i) &= f^\star(b_i, g_i) f^\star(Y_i|b_i, g_i) f^\star(T_i, \Delta_i|Y_i, b_i, g_i) \\ &= f^\star(b_i, g_i) f^\star(Y_i|b_i, g_i) f^\star(T_i, \Delta_i|Y_i, g_i) \qquad \text{(by Assumption 2)} \\ &= f^\star(b_i) f^\star(g_i) f^\star(Y_i|b_i, g_i) f^\star(T_i, \Delta_i|Y_i, g_i). \qquad \text{(by Assumption 3)} \end{aligned}$$

The negative complete log-likelihood is then given by

$$\mathcal{L}_n^{\text{comp}}(\theta) = -n^{-1} \sum_{i=1}^{n} \Big(\log f_\theta(b_i) + \sum_{k=1}^{K} \mathbb{1}_{\{g_i=k\}} \big(\log \mathbb{P}_\theta(g_i = k) + \log f_\theta(Y_i \,|\, b_i, g_i = k) + \log f_\theta(T_i, \Delta_i \,|\, Y_i, g_i = k) \big) \Big), \tag{3.7}$$

where $f_\theta(b_i)$ is the density of a multivariate gaussian $\mathcal{N}(0, D)$ distribution and $f_\theta(Y_i \,|\, b_i, g_i = k)$ is typically the density of a $\mathcal{N}\big(M_{ik}, \Sigma_i\big)$ distribution.

3.3.2 Penalized objective

To avoid overfitting and provide interpretation on which longitudinal markers are relevant for predicting time-to-event, we propose to minimize the penalized negative log-likelihood

$$\mathcal{L}_n^{\text{pen}}(\theta) = \mathcal{L}_n(\theta) + \Omega(\theta) = \mathcal{L}_n(\theta) + \sum_{k=1}^{K} \zeta_{1,k} \Omega_1(\xi_k) + \sum_{k=1}^{K} \zeta_{2,k} \Omega_2(\gamma_k), \tag{3.8}$$

where Ω_1 is an elastic net regularization (Zou and Hastie, 2005), Ω_2 is a sparse group lasso regularization (Simon et al., 2013), and $(\zeta_{1,k}, \zeta_{2,k})^\top \in (\mathbb{R}^+)^2$ regularization hyperparameters that need to be tuned. An advantage of this regularisation strategy is its ability to perform feature selection and to identify the most important features (longitudinal markers and time-independent) relative to the prediction objective. On the one hand, the support of ξ_k, controlled by the ℓ_1 term in Ω_1, provides information about the time-independent features involved in the k-th latent class membership while the ℓ_2 regularization allows to handle correlations between time-independent features. On the other hand, for the sparse group lasso penalty, a group ℓ corresponds to a trajectory, i.e. a longitudinal marker. Thus, if γ_k^ℓ is completely zero (thanks to the group lasso part), it means that the ℓ-th longitudinal process is discarded by the model in terms of risk effect for the k-th latent class. Then, the sparse part of the penalty allows a selection of association features for each trajectory: for γ_k^ℓ that are not completely zeroed, their support informs about the association features involved in the risk of the k-th latent class event for the ℓ-th longitudinal marker.

3.3.3 Optimization

Given our regularization strategy, we employ an extended version of the EM algorithm (McLachlan and Krishnan, 2007) which we now briefly outline. Extensive details on the algorithm are given in Section A.1 of the Supplementary Materials.

Our final optimization problem writes

$$\hat{\theta} \in \underset{\theta \in \mathbb{R}^P}{\arg\min}\, \mathcal{L}_n^{\text{pen}}(\theta). \tag{3.9}$$

Assume that we are at step $w+1$ of the EM algorithm, with current iterate denoted by $\theta^{(w)}$, then the algorithm consists in the following two steps:

- E-step: compute the expected negative complete log-likelihood conditional on the current estimate of the parameters $\theta^{(w)}$, that is, $\mathcal{Q}_n(\theta, \theta^{(w)}) = \mathbb{E}_{\theta^{(w)}}[\mathcal{L}_n^{\text{comp}}(\theta) \mid \mathcal{D}_n]$.
- M-step: find $\theta^{(w+1)} \in \underset{\theta \in \mathbb{R}^P}{\arg\min}\, \mathcal{Q}_n^{\text{pen}}(\theta, \theta^{(w)})$, where $\mathcal{Q}_n^{\text{pen}}(\theta, \theta^{(w)}) = \mathcal{Q}_n(\theta, \theta^{(w)}) + \Omega(\theta)$ and $\Omega(\theta)$ is the penalization defined in (3.8).

Under our assumptions, we can show that computing $\mathcal{Q}_n(\theta, \theta^{(w)})$ reduces to computing the expectations $\mathbb{E}_{\theta^{(w)}}[b_i | T_i, \Delta_i, Y_i]$ and $\mathbb{E}_{\theta^{(w)}}[b_i b_i^\top | T_i, \Delta_i, Y_i]$, and the probabilities $\mathbb{P}_{\theta^{(w)}}[g_i = k | T_i, \Delta_i, Y_i]$, $k \in \{1, \dots, K\}$, see Section A.1.1 of the Supplementary Materials for their exact expressions.

Concerning the M-step, we divide the problem into several updates for which we minimize $\mathcal{Q}_n^{\text{pen}}(\theta, \theta^{(w)})$ with respect to blocks of coordinates of θ separately. The order of these updates matters. The updates for D, $(\beta_k)_{k \in \{1,\dots,K\}}$, λ_0, and ϕ are easily obtained in closed-form. The update for $\xi_k^{(w)}$ reduces to the non-smooth convex minimization problem

$$\xi_k^{(w+1)} \in \underset{\xi \in \mathbb{R}^p}{\arg\min}\, \mathcal{F}_{1,k}(\xi) + \zeta_{1,k} \Omega_1(\xi), \tag{3.10}$$

where $\mathcal{F}_{1,k}$ is a convex function with respect to ξ. Problem (3.10) is then solved using a quasi-Newton method, the L-BFGS-B algorithm (Zhu et al., 1997). The update for $\gamma_k^{(w)}$ has a similar expression and is solved using proximal gradient descent (Boyd and Vandenberghe, 2004). We refer to Section A.1 of the Supplementary Materials for all details and proofs. The final algorithm is also given with a discussion of its convergence properties.

3.4 Evaluation methodology

In this section, we present our evaluation strategy to assess the real-time prediction performance of our model and briefly introduce the models used for comparison.

3.4.1 Real-time prediction and evaluation strategy

Developments in joint models have focused primarily on modeling and estimation, and most studies do not consider goodness-of-fit or predictive performance of latent class membership or time-to-event (Hickey et al., 2016). However, for real-time or daily predictions, practitioners need predictive prognostic tools to evaluate and compare survival models. Therefore, we place ourselves in a so-called "real-time" prediction setting. Once the learning phase for the model has been completed on a training set, so that one obtains $\hat{\theta}$ from (3.9) using the approach described in Section 3.3.3, we want to make real-time predictions. More precisely, for each subject i, we seek to provide a predictive marker, typically the probability of belonging to a latent class at any time t, using all the data available up to that time, but without using the supervision labels (T_i, Δ_i), which are a priori not available at any time t.

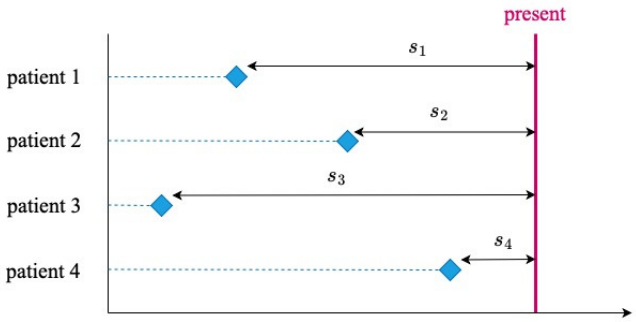

Figure 3.4 – Real-time prediction setting. In a practical application, we want to be able to make predictions at any "present" time while subjects have entered the study at different times. Therefore, some of them have a lot of recorded information while the others have a few.

Predictive marker

In our setting, since each latent class represents the different risk levels of a subject, we choose the probability of latent class membership as the predictive marker. This is similar to what is classically done in JLCMs, where $\widetilde{\pi}_{ik}^{\hat{\theta}} = \mathbb{P}_{\hat{\theta}}[g_i = k | T_i, \Delta_i, Y_i]$ is typically used as the predictive rule (see, e.g., Proust-Lima et al., 2014). However, this requires knowledge of the survival labels (T_i, Δ_i), which does not fit in our real-time prediction goal. Therefore, we define a new predictive marker as follows.

For any subject i and any time s_i elapsed since entry into the study, given longitudinal markers $\mathcal{Y}_i(s_i^-)$ observed up to s_i, for any $k \in \{1, \dots, K\}$, we let

$$\widehat{\mathcal{R}}_{ik}(s_i) = \mathbb{P}_{\hat{\theta}}\big[g_i = k \,|\, T_i^\star > s_i, \mathcal{Y}_i(s_i^-)\big].$$

Indeed, for any subject i who is event-free when s_i has elapsed, all we know about that subject is that its time to the event of interest $T_i^\star$ exceeds s_i. This is equivalent to considering this subject as a new subject for which $T_i = s_i$, $\Delta_i = 0$, and $Y_i = \mathcal{Y}_i(s_i^-)$. The expression of $\widehat{\mathcal{R}}_{ik}(s_i)$ can thus be rewritten as $\widehat{\mathcal{R}}_{ik}(s_i) = \mathbb{P}_{\hat{\theta}}[g_i = k \,|\, T_i = s_i, \Delta_i = 0, Y_i = \mathcal{Y}_i(s_i^-)]$, see Lemma 3 of the Supplementary Materials for more details.

We illustrate this real-time prediction setting in Figure 3.4, emphasizing that s_i should be thought of as the duration between the enrollment of individual i and the "present" time.

Performance evaluation

We want to compare the quality of our predictions to the true labels (T_i, Δ_i), to which we have access in these comparison experiments. Given a test set in which each individual's trajectory is fully observed until the end of the study, we mimic the real-time prediction setting by randomly sampling the s_i.

We use the classical C-index (Harrell et al., 1996) as our performance metric. More precisely, we assume that we are in the case $K = 2$ and the class $g_i = 2$ represents the

high-risk group of subjects (the class $g_i = 1$ then representing a low-risk group). We denote by $\widehat{\mathcal{R}}_i = \widehat{\mathcal{R}}_{i2}(s_i)$ the predictive marker that a subject i belongs to class $g_i = 2$ when s_i has elapsed. Then, we let $\mathcal{C} = \mathbb{P}[\widehat{\mathcal{R}}_i > \widehat{\mathcal{R}}_j \,|\, T_i^\star < T_j^\star]$, with $i \neq j$ two random independent subjects (note that $\mathcal{C}$ does not depend on i, j under the i.i.d. sample hypothesis).

In our case, $T^\star$ is subject to right censoring, so one would typically consider the modified $\widetilde{\mathcal{C}}$ defined by $\widetilde{\mathcal{C}} = \mathbb{P}[\widehat{\mathcal{R}}_i > \widehat{\mathcal{R}}_j \,|\, T_i < T_j, T_i < t^{\max}]$, where $t^{\max}$ corresponds to a fixed and predetermined follow-up period (Heagerty and Zheng, 2005). It has been shown by Uno et al. (2011) that a Kaplan-Meier estimator for the censoring distribution leads to a nonparametric and consistent estimator of $\widetilde{\mathcal{C}}$.

In Section A.3.4 of the Supplementary Materials, we give the complete procedure used to evaluate the performance of the models considered in our experiments.

3.4.2 Competing models

We compare FLASH with the very classical and widely used `LCMM` (Proust-Lima et al., 2015) and `JMbayes` Rizopoulos (2016a), which are extensions of JLCM and SREM that allow for multivariate longitudinal markers. We present them, along with their respective predictive markers, in Section A.2 of the Supplementary Materials. Note that not many joint models allow for multivariate longitudinal markers, which limits our choice of competing methods.

3.5 Experimental results

To evaluate our method, we first perform in Subsection 3.5.1 a simulation study that illustrates our estimation procedure. We then turn to a comparison study on both simulated and medical examples in Subsection 3.5.2, and finally show in Subsection 3.5.3 that the biomarkers identified as significant by our model are consistent with current medical knowledge.

In all experiments, the features extracted by the `tsfresh` package (Christ et al., 2018) are used for association features Ψ_m in FLASH. This package extracts dozens of features from a time series such as absolute energy, kurtosis, or autocorrelation. Before running our extended EM algorithm with the set of features extracted by the `tsfresh` package, we use a screening phase procedure where we select the top ten association features by fitting the extracted feature of each candidate and the survival labels in individual Cox models and comparing their C-index scores. In addition, a recent line of work is to use the signature transform (Fermanian, 2021; Bleistein et al., 2024) to extract features from longitudinal markers. This transform encapsulates geometric information about multivariate time series. We provide additional results with the signature transform in Section A.3.7 of the Supplementary Materials, which show that our method is generic and performs well regardless of the feature extraction functions used.

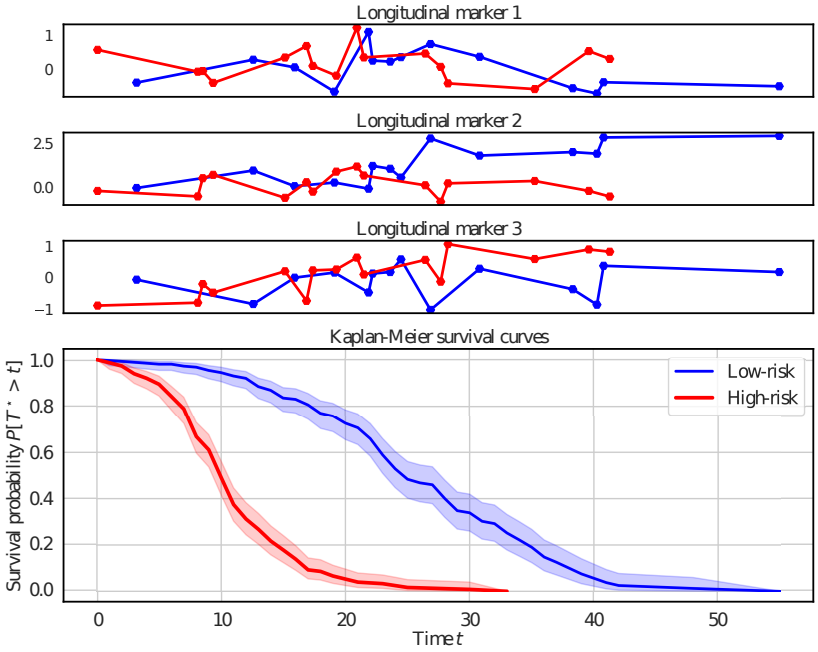

Figure 3.5 – Simulated cohort of $n = 500$ samples for $K = 2$ groups (high-risk group in red curves and low-risk group in blue curves). Top figures: trajectories of first three longitudinal markers of two individuals randomly selected in each group. Bottom figure: Kaplan-Meier survival curves for each group.

We tune the regularization hyperparameters $(\zeta_{1,k}, \zeta_{2,k})_{k\in\{1,\dots,K\}}$ with a grid search and a 10-fold cross-validation with the C-index metric. Note that we keep $\zeta_{1,1} = \cdots = \zeta_{1,K}$ and $\zeta_{2,1} = \cdots = \zeta_{2,K}$. Extensive details on our experiments together with additional results on a high-dimensional dataset from NASA are given in Section A.3 of the Supplementary Materials.

3.5.1 Simulation study

To assess our estimation procedure, we simulate data as follows. First, we simulate the latent class membership indicator from the logistic regression model in (3.1). Based on this indicator, we divide the population into two groups: a high-risk group and a low-risk group. Within each group, we apply the classical survival simulation setting described by Bender et al., 2005. Next, we generate the longitudinal markers Y from the generalized linear mixed models in (3.3). Survival times are generated from their hazard functions given in (3.4). The model coefficients ξ_1, ξ_2, and γ of models (3.1) and (3.4) are generated as sparse vectors, ensuring that only a subset of the corresponding features are active (i.e., the coefficients are non-zero). Extensive details of these simulations are provided in Section A.3.2 of the Supplementary Materials. Figure 3.5 shows some examples of simulated longitudinal markers and Kaplan-Meier survival curves.

To illustrate our regularization strategy, we show in Figure 3.6 the time-independent parameter ξ and the joint association parameters $(\gamma_k)_{k\in\{1,2\}}$ and their estimation obtained

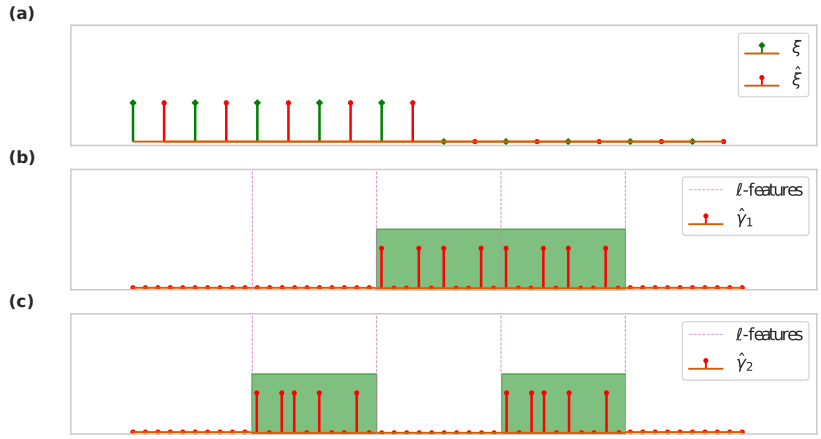

Figure 3.6 – Simulations results. **(a)**: the support of both the true coefficient ξ in green and its estimated version $\hat{\xi}$ in red. **(b)** and **(c)**: in red the support of the estimated coefficient $\hat{\gamma}_k$ for $k \in \{1, 2\}$, the dashed pink lines separate the features corresponding to each longitudinal marker ℓ, and active longitudinal markers are represented by a green area.

after running our learning procedure. We see in the sub-figure **(a)** that the support of ξ is fully recovered thanks to the elastic-net penalty. Additionally, sub-figures **(b)** and **(c)** demonstrate the effect of the sparse group lasso, showing that only the coefficients corresponding to active longitudinal features (represented by the green area) are non-zero, while all coefficients for inactive longitudinal features are zero.

3.5.2 Comparison study

We compare FLASH with `JMbayes` and `LCMM` on two simulated and two real-world datasets and use the C-index metric presented in Section 3.4. The first simulated dataset is the one from the previous subsection, and the second one is from the R package (Hickey et al., 2018) `joineRML`. We give detailed description of this second simulated dataset and the two real-world datasets (*PBCseq* and *Aids*) in Section A.3.3 of the Supplementary Materials. A summary of the datasets is given in Table 3.1.

We can see in Figure 3.7 that FLASH outperforms its competitors in terms of both C-index and running times on all datasets. The good performance of FLASH in terms of running times can be explained by the fact that it does not need to perform computationally intensive Monte Carlo techniques like `JMbayes`, while it is easier to satisfy the convergence criterion of our EM algorithm than that of `LCMM`.

3.5.3 Biological interpretation of FLASH results

Considering the growing emphasis on model interpretability and the fact that the new regulations in the European Union and the United States now require that a model be interpretable and understandable to be certified as a medical device (Geller, 2023; Panigutti et al., 2023), we conclude this section with interpretations of the results of FLASH on the PBCseq

Table 3.1 – Datasets characteristics: the number of samples n, the number of longitudinal features L, number of time-independent features p, and the overall number of parameters in FLASH model P. The names FLASH_simu and joineRML_simu correspond to the datasets simulated from the simulation study in Section 3.5.1 and the `joineRML` package respectively.

Dataset	n	L	p	P
FLASH_simu	500	5	10	224
joineRML_simu	250	2	2	204
PBCseq	304	7	3	251
Aids	467	1	4	147
Sepsis	654	4	21	255

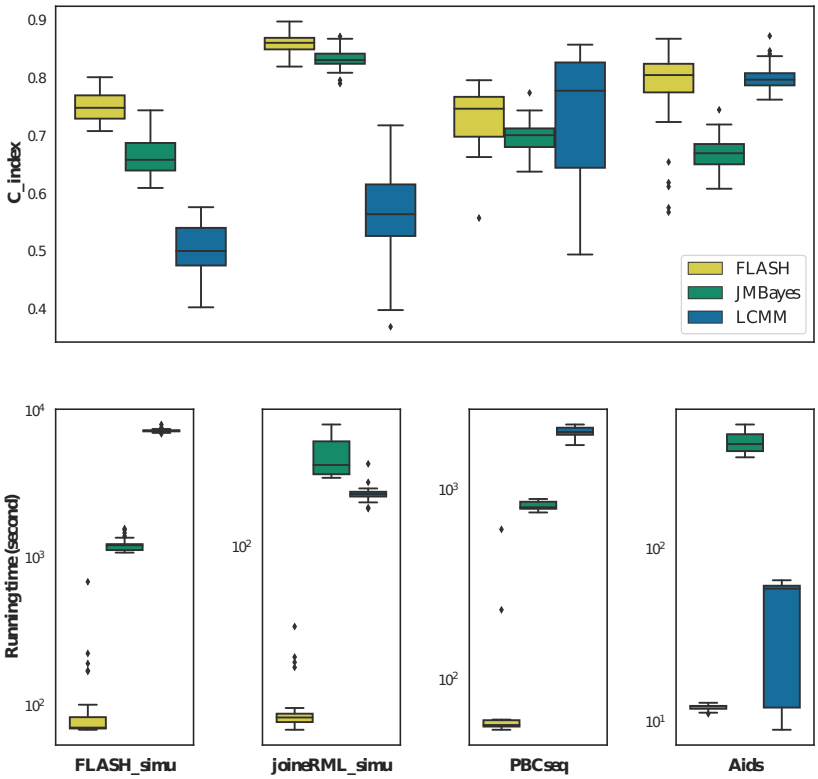

Figure 3.7 – C-index (**top** figure) and runtime (**bottom** figures) comparison on the four datasets considered. The box plots of C-index and runtime are obtained with 50 independent experiments.

and Sepsis datasets. The Sepsis dataset describes the sepsis diagnosis of patients, where, after a pre-processing step, 4 multivariate longitudinal features and 21 time-independent features are available for each patient. The coefficients estimated by FLASH, and in particular their sparsity, provide us information on which marker is involved in the diagnosis, see Section A.3.5 of the Supplementary Materials for all numerical values. Note that we did not include the Sepsis dataset in Section 3.5.2 because both `LCMM` and `JMBayes` take

a very long time to converge, so that we could not do the Monte Carlo comparison. This highlights the fact that they do not scale to high-dimensional settings. However, on one experimental trial, `LCMM` and `JMBayes` only give C-index scores of 0.51 and 0.56 while FLASH has a score of 0.74.

For the PBCseq dataset, alkaline phosphatase appears to be the most important variable, followed by prothrombin and albumin. Alkanine phosphatase is already recognized to be an important variable to monitor: it is already known that phosphatase alkaline at 6-month predicts non-responders and survival (Perez et al., 2023), it is recognized that treatment target should be normalization of alkaline phosphatase (Perez et al., 2020), and prognosis is improved for patients taking drugs lowering alkaline phosphatase as ursodeoxycholic acid (Kuiper et al., 2009). Concerning the Sepsis dataset, respiratory rate is the most important variable. Systolic blood pressure appears to be the most important longitudinal feature, followed by oxygen saturation. Of note, the two most widely used prognostic criteria in sepsis, i.e. the qSOFA and the SIRS criteria, which contain respectively 3 and 4 variables, both include among these variables respiratory rate (Raith et al., 2017).

3.6 Discussion

In this paper, a generalized joint model for high-dimensional multivariate longitudinal data and censored durations (FLASH) has been introduced, with an efficient estimation methodology based on an extension of the EM algorithm. This algorithm allows the use of regularization strategies in order to perform feature selection and results in an interpretable model scalable to high-dimensional longitudinal markers. We evaluated the performance of the estimation procedure on an extensive Monte Carlo simulation study. It showed that our method successfully recovered the most significant features. The proposed methodology has then been applied on four different datasets. On these datasets, FLASH outperforms competing methods, both in terms of C-index and runtimes, in a so-called "real-time" prediction setting. In addition, we show on experiments of medical datasets that our model automatically identifies the most important longitudinal markers and time-independent features, allowing important interpretations on the application at hand. Potential future work consists of extending the implementation to generalize our EM algorithm to support count or binary longitudinal features, and to relax the assumptions on latent class membership to allow classes to change with time, using for example a Markov structure.

Chapter 4

Dynamic Survival Analysis with Controlled Latent States

In this chapter, my main contribution is code implementation and experiments.

Contents

4.1 Introduction

Time-to-event data is ubiquitous in numerous fields such as meteorology, economics, healthcare, and finance. We typically want to predict when an event - which can be a catastrophic earthquake, the burst of a housing bubble, the onset of a disease, or a financial crash - will occur by using some prior historical information (Ogata, 1988; Bacry et al., 2015; Bussy et al., 2019b). This general problem encompasses many settings and in particular survival analysis, where every individual experiences at most one event (Cox, 1972b).

For an individual i, we have access to several event times $T_1^i < T_2^i < \dots$ and features $\mathbf{W}^i \in \mathbb{R}^s$ measured at time 0. For instance, in neurology, one might consider the onset times of a series of seizures (Rasheed et al., 2020) and $\mathbf{W}^i$ summarizes unchanging characteristics of the individual (age, gender, ethnicity, ...). The physician's goal is to determine whether an individual has a high probability of experiencing a seizure at time t given their characteristics. Such a task is most often addressed by modeling the individual specific intensity of a counting process of the form $\sum_{j\geq 1} \mathbb{1}_{T_j^i \leq t}$, using, for instance, Cox models (Cox, 1972b; Aalen et al., 2008; Kvamme et al., 2019) or Hawkes processes in the case of self-exciting processes (Bacry et al., 2015). Recent advances in the field have also enriched these models using deep architectures (Mei and Eisner, 2017; Kvamme et al., 2019; Omi et al., 2019; Chen et al., 2021; Groha et al., 2020; Shchur et al., 2021; De Brouwer et al., 2022; Tang et al., 2022). Once learnt, the intensity of the process can be used to predict occurrence times of future events or rank individuals based on their relative risks.

Learning with Time-dependent Data. More realistically, in addition to the static features $\mathbf{W}^i$, we often have access to time-dependent features along with their sampling times

$$\mathbf{X}^i =: \{(\mathbf{X}^i(t_1), t_1), \dots, (\mathbf{X}^i(t_K), t_K)\} \in \mathbb{R}^{d\times K},$$

where $D = \{t_1, \dots, t_K\} \subset [0, \tau]$ is a set of measurement times and τ the end of study. Retaking the example of seizure prediction, the time-dependent features may represent some measurements made by a wearable device, as done for instance by Dumanis et al. (2017). Taking both the static and time-dependent information into account is crucial when making predictions. This setting calls for highly flexible models of the intensity which take into account the stream of information carried by the longitudinal features.

From joint models to ODE-based methods. This problem has been tackled by the biostatistics community, in particular using joint models that concurrently fit parametric models to the trajectory of the longitudinal features and the intensity of the counting process (Ibrahim et al., 2010; Crowther et al., 2013; Proust-Lima et al., 2014; Long and Mills, 2018). Popular implementations include `JMBayses` (Rizopoulos, 2016b). While being highly interpretable, they do not scale to high-dimensional and frequently measured data, despite some recent algorithmic advances (Hickey et al., 2016; Murray and Philipson, 2022; Rustand et al., 2024) adapted to moderate dimension (up to $\simeq 5$ longitudinal features).

Modern deep methods, that can encode complex and meaningful patterns from complex

data in latent states, offer a particularly attractive alternative for this problem. However, the literature bridging the gap between deep learning and survival analysis is scarce. Notably, Lee et al. (2019) tackle this problem by embedding the time-dependent data through a recurrent neural network combined with an attention mechanism. They then use this embedding in a discrete-time setting to maximize the likelihood of dying in a given time-frame conditional on having survived until this time. Moon et al. (2022) combine a probabilistic model with a continuous-time neural network, namely the ODE-RNNS of Rubanova et al. (2019) in a similar setup.

Modelling Time Series with Controlled Latent States. Building on the increasing momentum of differential equation-based methods for learning (Chen et al., 2018; De Brouwer et al., 2019; Rubanova et al., 2019; Chen et al., 2021; Moon et al., 2022; Marion et al., 2022), we propose a novel modelling framework in which the unknown intensity of the counting process is parameterized by a latent state driven by a controlled differential equation (CDE). Formally, we let the unknown intensity of the counting process of individual i depend on their covariates $\mathbf{W}^i$ and an unobserved process $x^i : [0, \tau] \to \mathbb{R}^d$ that is the continuous unobserved counterpart of the time series $\mathbf{X}^i$, i.e., $(\mathbf{X}^i(t), t) = x^i(t)$ for all $t \in D$. We model the intensity (i.e. the instantaneous probability of experiencing an event — see Section 4.2.2) by setting

$$\lambda^i_\star\big(t \,|\, \mathbf{W}^i, (x^i(s))_{s\le t}\big) = \exp\big(z^i_\star(t) + \beta_\star^\top \mathbf{W}^i\big), \tag{4.1}$$

where the dynamical latent state $z^i_\star(t) \in \mathbb{R}$ is the solution to the CDE

$$dz^i_\star(t) = \mathbf{G}_\star\big(z^i_\star(t)\big)^\top dx^i(t) \tag{4.2}$$

with initial condition $z^i_\star(0) = 0$ driven by x^i. Here, the vector field $\mathbf{G}_\star : \mathbb{R} \to \mathbb{R}^d$ and $\beta_\star \in \mathbb{R}^s$ are both unknown. This means that the latent dynamics are common between individuals, but are driven by individual-specific data, yielding individual-specific intensities. Such a modelling strategy is reminiscent of state space models, which embed times series through linear controlled latent differential equations (Gu et al., 2022; Cirone et al., 2024). Our framework is introduced in more detail later.

Contributions. In an effort to provide scalable and efficient models for event-data analysis, we propose two novel estimators. We first leverage neural CDEs (Kidger et al., 2020), which directly approximate the vector field $\mathbf{G}_\star$ with a neural vector field $\mathbf{G}_\psi$. In a second time, following Fermanian et al. (2021) and Bleistein et al. (2023), we propose to linearize the unknown dynamic latent state $z^i_\star(\cdot)$ in the signature space. Informally, this means that at any time t, we have the simplified expression

$$z^i_\star(t) \approx \alpha^\top_{\star,N} \mathbf{S}_N(x^i_{[0,t]})$$

where $\alpha_{\star,N}$ is an unknown finite-dimensional vector and $\mathbf{S}_N(x^i_{[0,t]})$ is a deterministic transformation of the time series x^i observed up to time t called the *signature transform*. Notice that in this form, the vector $\alpha_{\star,N}$ does not depend on t and can hence be learned at any observation time. We obtain theoretical guarantees for both models ; for the second model

in particular, we state a precise decomposition of the variance and the discretization bias of our estimator, which crucially depends on the coarseness of the sampling grid D. Finally, we benchmark both methods on simulated and real-world datasets from finance, healthcare and digital food retail, in a survival analysis setting. Our signature-based estimator provides state-of-the-art results.

Summary. Section 4.2 details our theoretical framework. In Section 4.3, we state theoretical guarantees for our model. Lastly, we conduct a series of experiments in Section 4.4 that displays the strong performances of our models against an array of benchmarks. All proofs are given in the appendix. The code is available at *https://github.com/LinusBleistein/signature_survival*.

4.2 Modelling Point Processes with Controlled Latent States

4.2.1 The Data

In practice, an individual can be censored (for example after dropping out from a study) or cannot experience more than a given number of events. To take this into account, we introduce $Y^i : [0,\tau] \to \{0,1\}$ the at-risk indicator function, which equals 1 when the individual i is still at risk of experiencing an event. Together with Y^i, we define

$$N^i(t) := \sum_{j\geq 1} \mathbb{1}_{T_j^i \leq t} Y^i(T_j^i)$$

as the stochastic process counting the number of events experienced by individual i up to time t and while $Y^i(T_j^i) = 1$. Our dataset

$$\mathcal{D}_n := \{\mathbf{X}^i, \mathbf{W}^i, Y^i(t), N^i(t),\, 0 \leq t \leq \tau\}$$

consists of n i.i.d. historical observations up to time τ. Our setup can be extended to individual-dependent grids $(D^i)_{i=1}^n$, but we choose to focus on the former setting for the sake of clarity. The individual specific time series are only observed as long as the individual is at risk. We first make an assumption on the time series.

Assumption 4. For every individual $i = 1, \dots, n$, there exists a continuous path of bounded variation $x^i : [0,\tau] \to \mathbb{R}^d$ satisfying, for all $0 \leq s < t \leq \tau$,

$$\left\|x^i\right\|_{1\text{-var},[s,t]} := \sup_D \sum_k \left\|x^i(t_{k+1}) - x^i(t_k)\right\| \leq L_x|t-s|$$

where $\|\cdot\|$ is the Euclidean norm and the supremum is taken over all finite dissections $D = \{s = t_1 < \cdots < t_K = t\}$. The time series $\mathbf{X}^i$ is a discretization of x^i on the grid D.

Remark that this assumption implies that the paths are L_x-Lipschitz. We now state a supplementary assumption on the static features.

Assumption 5. There exists a constant $B_{\mathbf{W}} > 0$ such that for every $i = 1, \dots, n$, $\|\mathbf{W}^i\|_2 \leq B_{\mathbf{W}}$.

4.2.2 Modelling Intensities with Controlled Differential Equations

Intensity of a counting process. We define the individual-specific intensity $\lambda_\star^i\big(t \mid \mathbf{W}^i, x^i_{[0,t]}\big)$ of the underlying counting process, which we will simply write $\lambda_\star^i(t)$ in the following, as

$$\lambda_\star^i(t) := \lim_{h \to 0^+} \frac{1}{h} \mathbb{E}\big(N^i(t+h) - N^i(t) \,|\, \mathcal{F}_t^i\big)$$

where $\mathcal{F}_t^i$ is the past information at time t which includes $\mathbf{W}^i$ and $x^i_{[0,t]}$ (Aalen et al., 2008).

Controlled Dynamics. Controlled differential equations are a theoretical framework that allows to generalize ODEs beyond the non-autonomous regime Lyons et al., 2007. Recall that a non-autonomous ODE is the solution to

$$dz(t) = \mathbf{F}(z(t), t)dt$$

with a given initial value $z(0) = z_0 \in \mathbb{R}^p$. Here, the vector field $\mathbf{F} : \mathbb{R}^p \times [0, +\infty[\to \mathbb{R}^p$ depends explicitly on $t \geq 0$, allowing for time-varying dynamics unlike autonomous ODEs whose dynamics remain unchanged through time. Controlled differential equations can be seen as a generalization of non-autonomous ODEs. They allow for the vector field to depend explicitly on the values of another $\mathbb{R}^d$-valued function $x : [0, 1] \to \mathbb{R}^d$ through

$$dz(t) = \tilde{\mathbf{F}}(z(t), x(t))dt$$

thus encoding even richer dynamics. Formally, a CDE writes

$$\begin{aligned} dz(t) &= \mathbf{G}\big(z(t)\big)dx(t) \\ z(0) &= z_0 \in \mathbb{R}^p \end{aligned}$$

where $\mathbf{G}$ is a $\mathbb{R}^{p \times d}$-valued vector field. Existence and uniqueness of the solution is ensured under regularity conditions on $\mathbf{G}$ and x by the Picard-Lindelhöf Theorem (see Theorem 2). The following assumption is needed in order to ensure that the function

$$\lambda_\star^i(t) = \exp\big(z_\star^i(t) + \beta_\star^\top \mathbf{W}^i\big),$$

where the dynamical latent state $z_\star^i(t) \in \mathbb{R}$ is the solution to the CDE

$$dz_\star^i(t) = \mathbf{G}_\star\big(z_\star^i(t)\big)^\top dx^i(t)$$

with initial condition $z_\star^i(0) = 0$ driven by x^i is well-defined.

Assumption 6. The vector field $\mathbf{G}_\star : \mathbb{R} \to \mathbb{R}^d$ defining $\lambda^i_\star$ in Equation (4.2) is $L_{\mathbf{G}_\star}$-Lipschitz; $\beta_\star$ is such that $\|\beta_\star\|_2 \leq B_{\beta,2}$, $\|\beta_\star\|_1 \leq B_{\beta,1}$ and $\|\beta_\star\|_0 \leq B_{\beta,0}$, where $B_{\beta,2}, B_{\beta,1}, B_{\beta,0} > 0$ are constants.

Under these assumptions, the intensity is bounded at all times.

Lemma 1 (A bound on the intensity)**.** For every individual $i = 1, \dots, n$ and all $t \in [0, \tau]$, the log intensity $\log \lambda^i_\star(t)$ is upper bounded by

$$B_{\beta,2} B_{\mathbf{W}} + \|\mathbf{G}_\star(0)\|_{\text{op}} L_x t \exp\left(L_{\mathbf{G}_\star} L_x t\right)$$

almost surely.

This is a direct consequence of Lemma 3.3 in Bleistein and Guilloux (2024). Remark that $\|\mathbf{G}_\star(0)\|_{\text{op}} < \infty$ since the vector field is Lipschitz and hence continuous.

Remark 1. By differentiation, one can see that the intensity itself satisfies a so-called controlled Volterra differential equation (Lin and Yong, 2020). Indeed, differentiating the intensity $\lambda^i_\star$ yields the CDE

$$d\lambda^i_\star(t) = \lambda^i_\star(t)\mathbf{G}_\star(z^i_\star(t))dx^i(t)$$

with initial condition $\lambda^i_\star(0) = \exp(\beta_\star^\top \mathbf{W}^i)$. Note that this CDE is path dependent, i.e., its vector field depends on the path $z^i_\star : [0, \tau] \to \mathbb{R}$.

Remark 2. This model enforces continuity of the intensity: indeed, the solution of a CDE inherits the regularity of its driving path. A possible solution to accommodate discontinuous intensity functions is to add a jump term to the generative CDE, which could then be learnt using neural jump ODEs (Jia and Benson, 2019).

4.2.3 Neural Controlled Differential Equations

Following the ideas of continuous time models, our first approach to learning the dynamics is to fit a parameterized intensity to this model by setting

$$\lambda^i_\theta(t) = \exp(\alpha^\top z^i_\theta(t) + \beta^\top \mathbf{W}^i),$$

where $z^i_\theta(t) \in \mathbb{R}^p$ is an embedding of the time series $\mathbf{X}^i$ parameterized by $\theta \in \mathbb{R}^v$ and $\alpha \in \mathbb{R}^p$ is a learnable parameter. We propose to use Neural Controlled Differential Equations (NCDEs), a powerful tool for embedding irregular time series introduced by Kidger et al. (2020). NCDEs work by first embedding a time series $\mathbf{X}^i$ in the space of functions of bounded variation, yielding $x^{i,D} : [0, \tau] \to \mathbb{R}^d$, before defining a representation of the data through

$$dz_\theta(t) = \mathbf{G}_\psi\big(z_\theta(t)\big)dx^{i,D}(t)$$

with initial condition $z_\theta(0) = \mathbf{0}$. It is common practice to set $\mathbf{G}_\psi : \mathbb{R}^p \to \mathbb{R}^{p\times d}$ to be a small feed-forward neural network parameterized by ψ. The learnable parameters of this model are thus $\theta = (\alpha, \psi, \beta)$. In our setup, the embedding must be carefully chosen in order not to leak information from the future observations. Hence natural cubic splines, used in the original paper by Kidger et al. (2020), cannot be used and we resort to the piecewise constant interpolation scheme proposed by Morrill et al. (2021) and defined as $x^{i,D}(s) = (\mathbf{X}^i(t_k), s)$ for all $s \in [t_k, t_{k+1}[$. This yields a discretely updated latent state equal to

$$z_\theta^{i,D}(t_k) = z_\theta^{i,D}(t_{k-1}) + \mathbf{G}_\psi(z_\theta^{i,D}(t_{k-1}))\Delta\mathbf{X}^i(t_k)$$

where $\Delta\mathbf{X}^i(t_k) = \mathbf{X}^i(t_k) - \mathbf{X}^i(t_{k-1})$. This architecture has been studied under the name of *controlled ResNet* because of its resemblance with the popular ResNet (Cirone et al., 2023; Bleistein and Guilloux, 2024).

In order to provide theoretical guarantees, we restrict ourselves to a bounded set of NCDEs i.e. we consider a set of NCDE predictors

$$\Theta_1 = \{\theta \in \mathbb{R}^v \text{s.t. } \|\alpha\|_2 \leq B_\alpha, \|\psi\| \leq B_\psi, \|\beta\|_2 \leq B_{\beta,2}\}$$

where the norm on ψ refers to the sum of ℓ_2 norms of the weights and biases of the neural vector field $\mathbf{G}_\psi$. This restriction is fairly classical in statistical learning theory Bach, 2021.

4.2.4 Linearizing CDEs in the Signature Space

The Signature Transform. While neural controlled differential equations allow for great flexibility in representation of the time series, they are difficult to train and require significant computational resources. The signature is a promising and theoretically well-grounded tool from stochastic analysis, that allows for a parameter-free embedding of the time series. Mathematically, the signature coefficient of a function

$$x : t \in [0, \tau] \mapsto \left(x^{(1)}(t), \ldots, x^{(d)}(t)\right)$$

associated to a word $I = (i_1, \ldots, i_k) \in \{1, \ldots, d\}^k$ of size k is the function

$$\mathbf{S}^I(x_{[0,t]}) := \int_{0<u_1<\cdots<u_k<t} dx^{(i_1)}(u_1) \ldots dx^{(i_k)}(u_k)$$

which maps $[0, \tau]$ to $\mathbb{R}$. The integral is to be understood as the Riemann-Stieltjes integral. While the definition of the signature is technical, it can simply be seen as a feature extraction step. We refer to Figure 4.1 for an illustration. The truncated signature of order $N \geq 1$, which we write $\mathbf{S}_N(x_{[0,t]})$, is equal to the collection of all signature coefficients associated to words of size $k \leq N$ sorted by lexicographical order. Finally, the infinite signature is the sequence defined through

$$\mathbf{S}(x_{[0,t]}) = \lim_{N\to+\infty} \mathbf{S}_N(x_{[0,t]}).$$

Learning with Signatures. Signatures are a prominent tool in stochastic analysis since the pioneering work of Chen (1958) and Lyons et al. (2007). They have recently found successful applications in statistics and machine learning as a feature representation for irregular time series (Kidger et al., 2019; Morrill et al., 2020; Fermanian, 2021; Salvi et al., 2021; Fermanian, 2022; Lyons and McLeod, 2022; Bleistein et al., 2023; Horvath et al., 2023) and a tool for analyzing residual neural networks in the infinite depth limit (Fermanian et al., 2021).

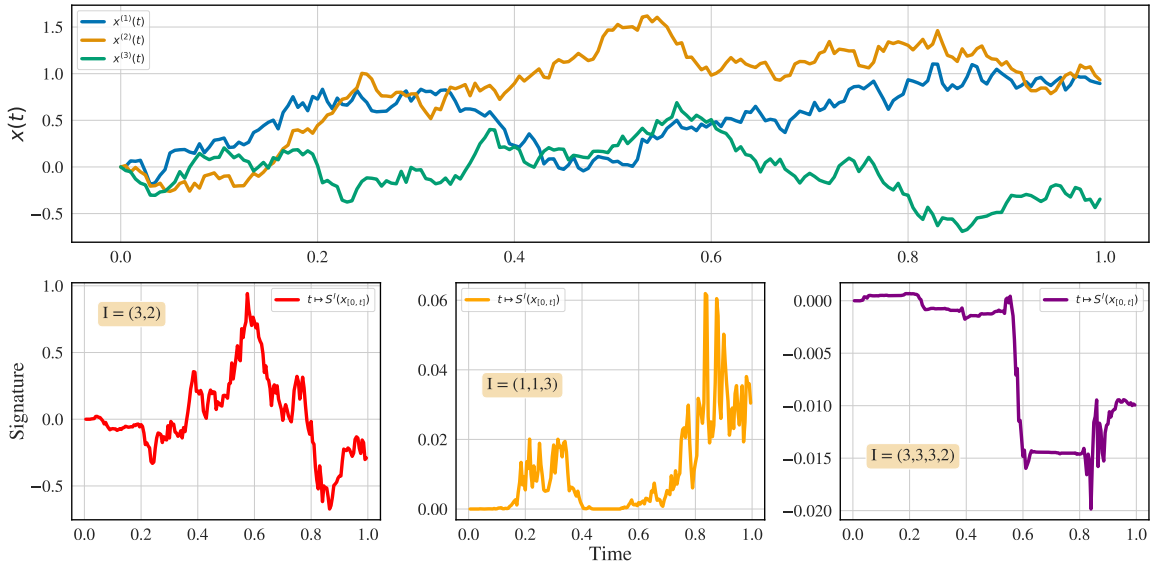

Figure 4.1 – Sample path $x(t)$ of a 3-dimensional fractional Brownian motion on **top**, and three signature coefficients $\mathbf{S}^I(x_{[0,t]})$ associated to different words on the **bottom**.

Signatures and CDEs. An appealing feature of signatures is their connection to controlled differential equations. Indeed, under sufficient regularity assumptions (Friz and Victoir, 2010; Fermanian et al., 2021; Bleistein et al., 2023; Cirone et al., 2023), the generative CDE (4.1) can be linearized in the signature space. Informally, this means that there exists a sequence $\alpha_\star$ such that for all $t \in [0, \tau]$ we have

$$z^i_\star(t) = \alpha_\star^\top \mathbf{S}(x^i_{[0,t]}).$$

The mathematical definition of $\alpha_\star$ is technical and we refer to Appendix B.1.4 for a formal statement and a discussion of the regularity assumptions. Hence, under the corresponding regularity conditions, the true intensity for individual i writes

$$\lambda^i_\star(t) = \exp\left(\alpha_\star^\top \mathbf{S}(x^i_{[0,t]}) + \beta_\star^\top \mathbf{W}^i\right).$$

This motivates the use of the signature-based estimator

$$\lambda^{i,D}_\theta(t) := \exp\left(\alpha^\top \mathbf{S}_N(x^{i,D}_{[0,t]}) + \beta^\top \mathbf{W}^i\right),$$

where $\theta = (\alpha, \beta) \in \mathbb{R}^q \times \mathbb{R}^s$, $N \geq 1$ is treated as a hyperparameter and $x^{i,D}$ corresponds to the piecewise constant embedding of the observed time series $\mathbf{X}^i$ described previously. The integer $q = \frac{d^{N-1}-1}{d-1}$ is the size of the signature truncated at depth $N \geq 1$. The superscript in D emphasizes the dependence of this estimator on the observation grid D. Similarly to the NCDE-based estimator, we restrict ourselves to the bounded set of estimators

$$\Theta_2 = \{\theta \text{ s.t. } \|\alpha\| \leq B_\alpha, \|\beta\| \leq B_{\beta,2}\}.$$

4.2.5 Connections to Cox Models with Time-Varying Covariates

Cox models with time-varying covariates are the classical class of models (Therneau and Grambsch, 2000; Aalen et al., 2008; Zhang et al., 2018), where the individual specific hazard rate has the form $\lambda^i_\theta(t) = \lambda_0(t) \exp(\alpha_\star^\top \mathbf{X}^i(t) + \beta_\star^\top \mathbf{W}^i)$, where $\lambda_0 : [0, \tau] \to \mathbb{R}_+$ is called the *baseline hazard*.

For signature-based embeddings, recall that we compute the signature of a time-embedded time series $\mathbf{X}^i = \{(\mathbf{X}^i(t_1), t_1), \ldots, (\mathbf{X}^i(t_k), t_k)\}$. In fact, this amounts to

$$\begin{aligned}&\alpha^\top \mathbf{S}_N(x^{i,D}_{[0,t]}) \\ &\quad= \underbrace{\sum_{k=0}^{N} \alpha_k t^k}_{=\log \lambda_0(t)} + \underbrace{\alpha_{I_1}^\top \mathbf{X}^i(t) + \sum_{I \in I_2} \alpha_I \mathbf{S}^I(x^{i,D}_{[0,t]})}_{=\log \text{ of individual specific hazard rate}},\end{aligned}$$

where α_{I_1} is a subvector of α and $I_2 \subset \prod_{k=2}^N \{1, \ldots, d\}^k$. Hence our model can be interpreted as a generalized version of Cox models with time-varying covariates. A similar interpretation holds for NCDEs. We detail this link in Appendix B.1.6.

4.3 Theoretical Guarantees

4.3.1 The Learning Problem

For both models, the parameter θ can be fitted by likelihood maximization by solving

$$\hat{\theta} \in \arg\min_{\theta \in \Theta} \ell_n^D(\theta) + \text{pen}(\theta), \tag{4.3}$$

where $\Theta \in \{\Theta_1, \Theta_2\}$ depending on whether one uses signature or NCDE-based embeddings, pen : $\Theta \to \mathbb{R}_+$ is a penalty and $\ell_n^D(\theta)$ is equal to the negative log-likelihood of the sample $\mathcal{D}_n$ evaluated at θ.

Unless specified other, the following statements hold for both NCDEs and signature-based embeddings (up to different constants given explicitly in the proofs). Following Aalen

et al. (2008), the negative log likelihood $\ell_n^D(\theta)$ of the sample writes

$$\frac{1}{n}\sum_{i=1}^{n}\int_0^\tau \lambda_\theta^{i,D}(s)Y^i(s)ds - \int_0^\tau \log\lambda_\theta^{i,D}(s)dN^i(s),$$

and we let

$$\ell_n^\star = \frac{1}{n}\sum_{i=1}^{n}\int \lambda_\star^i(s)Y^i(s)ds - \int \log\lambda_\star^i(s)dN^i(s)$$

be the true likelihood of the data. Our goal, in this section, is to obtain a bias-variance decomposition of the difference

$$\ell_n^D(\hat{\theta}) - \ell_n^\star$$

between the true likelihood and the likelihood of the learnt model.

4.3.2 A Risk Bound

Theorem 1 (Informal Risk Bound for the Signature Model)**.** Consider the signature-based embedding. Let $\hat{\theta}$ be the solution of (4.3) with $\text{pen}(\theta) = \eta_1 \|\alpha\|_1 + \eta_2 \|\beta\|_1$. For any $N \geq 1$, we have with high probability and an appropriate choice of η_1, η_2 that

$$\begin{aligned}\ell_n^D(\hat{\theta}) - \ell_n^\star \leq\ &\text{Discretization bias} + \text{Approximation bias}\\ &+ \mathcal{O}\left(\sqrt{\frac{\log N d^N}{n}}\right) + \mathcal{O}\left(\sqrt{\frac{\log s}{n}}\right).\end{aligned}$$

For a formal statement, see Appendix B.2. We make a series of comments on this result.

1. This full risk bound can only be obtained for the signature-based model. It can also be extended to other types of penalty such as Ridge or Group Lasso (see for instance Nardi and Rinaldo (2008)) For NCDEs, we are able to give precise guarantees on the bias following Bleistein and Guilloux (2024), but a precise control of the variance term is out of reach.
2. The discretization bias is proportional to $|D| := \max_{i=1,\dots,K} |t_i - t_{i-1}|$ and hence vanishes as sampling gets finer.
3. The approximation bias crucially depends on the regularity of the unknown tensor field $\mathbf{G}_\star$, and more precisely on the speed of decay of its derivatives, which can be seen as a measure of smoothness of the target function.
4. The regularity assumptions made on $\mathbf{G}_\star$ are not necessary to bound the approximation bias of the NCDE model: in this case, this bias term depends on the approximation capacities of the neural tensor field.
5. Remarkably, we obtain classical rates in $n^{-1/2}$ for the variance term. For signature based methods, fast rates in n^{-1} are yet to be obtained.

4.4 Experimental Evaluation

We now focus on the survival analysis setup. We hence let T^i be the unique time-of-event, which may eventually be censored, of individual i. Δ^i is the censorship indicator, equal to 1 if the individual experiences the event and to 0 otherwise.

4.4.1 Training Setup

We train on a dataset $\mathcal{D}_n$ of the same structure than described in Section 4.2.1 and learn the parameter $\hat{\theta}$ by solving the optimization problem (4.3). NCDEs are trained without penalization, while we use a mixture of elastic-net penalties

$$\text{pen}(\theta) := \eta_1 \text{pen}_{\text{EN}}(\alpha) + \eta_2 \text{pen}_{\text{EN}}(\beta)$$

for training the signature-based model, where $\text{pen}_{\text{EN}}(\cdot) = \gamma \|\cdot\|_1 + (1-\gamma)\|\cdot\|_2$. The hyperparameters (η_1, η_2, N) are chosen by cross-validation of a mixed metric equal to the difference between the C-index and the Brier score (see below) and we set $\gamma = 0.1$. We refer to Appendix B.3.1 for a detailed description of the training procedures. We evaluate our model's capacity to predict events in $[t, t+\delta t]$ by leveraging values of the longitudinal features up to t (see Figure 4.2) through a ranking metric and a calibration metric. This evaluation procedure is standard (Lee et al., 2019).

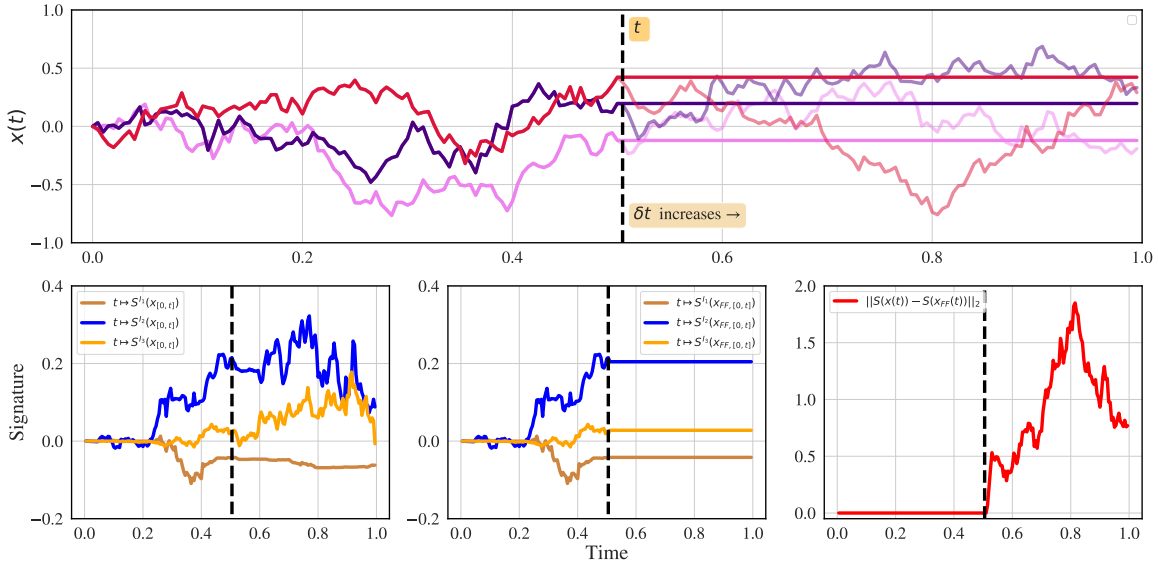

Figure 4.2 – On the **top**, observed time series up to time t in bold colors and true time series in faded colors. When evaluating our models, we fill-forward the last observed value from t on. On the **bottom**, signatures of the true path (**left**), of the observed path (**center**) and difference in ℓ_2 norm (**right**) — $x_{FF}(t)$ denotes the filled-forward time series.

4.4.2 Metrics

We compute four metrics using the individual specific survival functions as estimated by our model with parameters θ. At time $t + \delta t$ for $\delta t > 0$ conditional on survival up to time t, and on observation of the longitudinal features up to time t, it is defined as

$$r^i_\theta(t, \delta t) = \mathbb{P}\Big(T^i > t + \delta t \,|\, T^i > t, (\mathbf{X}^i(s))_{\substack{s \le t \\ s \in D}}, \mathbf{W}^i\Big).$$

We describe its detailed computation in Appendix B.3.2.

Time-dependent Concordance Index. Following Lee et al. (2019), we measure the discriminative power of our models by using a time-dependent concordance index $C(t, \delta t)$ that captures our models ability to correctly rank individuals on their predicted probability of survival. The concordance index $C(t, \delta t)$ is then finally computed as

$$\frac{\sum_{j=1}^{n}\sum_{i=1}^{n} \mathbb{1}_{r^i_\theta(t,\delta t) > r^j_\theta(t,\delta t)} \mathbb{1}_{T^i > T^j,\, T^j \in [t, t+\delta t], \Delta^j = 1}}{\sum_{j=1}^{n}\sum_{i=1}^{n} \mathbb{1}_{T^i > T^j,\, T^j \in [t, t+\delta t], \Delta^j = 1}}.$$

This metric captures the capacity of our model to discriminate between j and another individual i through the conditional probability of survival.

Brier Score. While the concordance index is a ranking-based measure, the Brier Score measures the accuracy in predictions by comparing the estimated survival function and the survival indicator function (Lee et al., 2019; Kvamme et al., 2019; Kvamme and Borgan, 2023). Formally, we define the Brier score $\mathrm{BS}(t, \delta t)$ as

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{T^i \le t+\delta t, \Delta^i = 1} r^i_\theta(t, \delta t)^2 + \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{T^i > t+\delta t}(1 - r^i_\theta(t, \delta t))^2.$$

Contrarily to the C-index, the Brier score is a measure of calibration of the predictions: it measures the distance between the estimated survival function and the indicator function of survival on the interval $[t, t + \delta t]$.

Averaged performance. Additionally, we evaluate the average prediction performance of our models over a set of different prediction times. The averaged C-index and Brier score on the interval $[t_1, t_2]$ along with the window time δt are defined respectively as

$$\frac{1}{t_2 - t_1}\int_{t_1}^{t_2} C(s, \delta t) ds \quad \text{and} \quad \frac{1}{t_2 - t_1}\int_{t_1}^{t_2} BS(s, \delta t) ds.$$

Comparison with static metrics. A crucial difference with static survival analysis metrics is that our metric only compares the individuals who experienced the event in this time window to all the ones who are still at risk at time t. This can lead to a C-index below 0.5 and Brier scores above 0.25 without the model being worse than random.

Additional metrics. We furthermore report AUC and weighted Brier score in Appendix B.4.

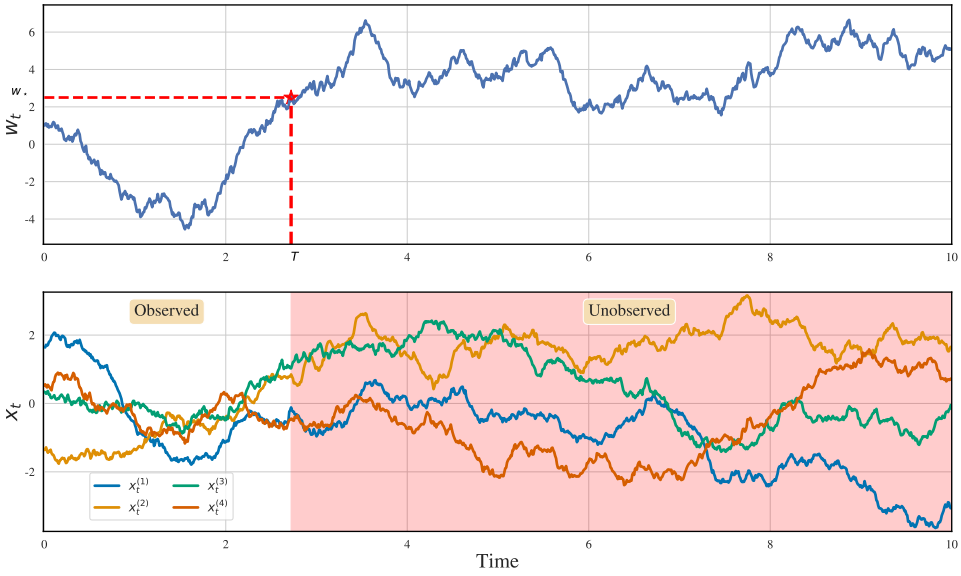

Figure 4.3 – Time series $\mathbf{X}^i$ of a randomly picked individual on **bottom** and unobserved SDE $w^i(t)$ on the **top**. The red star indicates the first hitting time of the threshold value $w_\star = 2.5$.

4.4.3 Methods

We propose three distinct methods. In addition to the signature-based model, which we call **CoxSig**, we also consider **CoxSig+** which adds the first value of the time series to the static features. This is motivated by the translation invariance of signatures (see discussion below). Our last method is the **NCDE** embedding of the longitudinal features. We benchmark our three models against a set of competing methods. All methods are detailed in Appendix B.3.1.

Time-Independent Cox Model. As a sanity check, we implement a simple Cox model with elastic-net penalty which uses the parameterized intensity $\lambda_\theta^i(t) = \lambda_0(t)\exp(\beta^\top \mathbf{W}^i)$ using `scikit-survival` (Pölsterl, 2020). This baseline allows to check whether our proposed methods can make use of the supplementary time-dependent information. If no

Name	n	d	Censoring	Avg. Times
Hitting time	500	5	Terminal (3.2%)	177
Tumor Growth	500	2	Terminal (8.4%)	250
Maintenance	200	17	Online (50%)	167
Churn	1043	14	Terminal (38.4%)	25

Table 4.1 – Description of the 4 datasets we consider. The integer d is the dimension of the time series including the time channel. *Terminal* censoring means that the individuals are censored at the end of the overall observation period $[0, \tau]$ if they have not experienced any event. It is opposed to *online* censoring that can happen at any time in $[0, \tau]$. The reported percentage indicates the censoring level i.e. the share of the population that does not experience the event. The last column reports the average number of observations times over individuals.

static features are available, we use the first observed value of the time series, i.e., $\mathbf{W}^i = \mathbf{X}^i(0)$.

Random Survival Forest (RSF). We use RSF (Ishwaran et al., 2008) with static features $\mathbf{W}^i$ as the only input. Similarly to our implementation of the Cox model, we use the first value of the time series as static features if no other features are available.

Dynamic DeepHit (Lee et al., 2019). DDH is a state-of-the-art method for dynamical survival analysis, that combines an RNN with an attention mechanism and uses both time dependent and static features.

SurvLatent ODE (Moon et al., 2022). SLODE is a recent deep learning framework for survival analysis that leverages an ODE-RNN architecture (Rubanova et al., 2019) to handle the time dependent features.

4.4.4 Synthetic Experiments

Hitting time of a partially observed SDE. Predicting hitting times is a crucial problem in finance — for instance, when pricing so-called catastrophe bounds triggering a payment to the holder in case of an event (Cheridito and Xu, 2015; Corcuera and Valdivia, 2016). Their relation to survival analysis is well documented, see e.g. Lee and Whitmore, 2006. Building on this problem, we consider the Ornstein-Uhlenbeck SDE

$$dw^i(t) = -\omega(w^i(t) - \mu)dt + \sum_{j=1}^{d} dx^{(i,j)}(t) + \sigma dB^i(t)$$

where $d = 5, \sigma = 1, \mu = 0.1$ and $\omega = 0.1$ are fixed parameters. $x^i(t) = (x^{(i,1)}(t), \dots, x^{(i,d-1)}(t))$ is a sample path of a fractional Brownian motion with Hurst parameter $H = 0.6$, and $B^i(t)$ is a Brownian noise term. In this setup, our data consists of $\mathbf{X}^i$ which is a downsampled

version of x^i and the Brownian part is unobserved. Our goal is to predict the first hitting time $\min\{t > 0 \,|\, w_t \geq w_\star\}$ of a threshold value $w_\star = 2.5$. We train on $n = 500$ individuals. Figure 4.3 shows the sample paths and SDE of a randomly selected individual. This setup is close to a well-specified model since signatures linearize controlled differential equations.

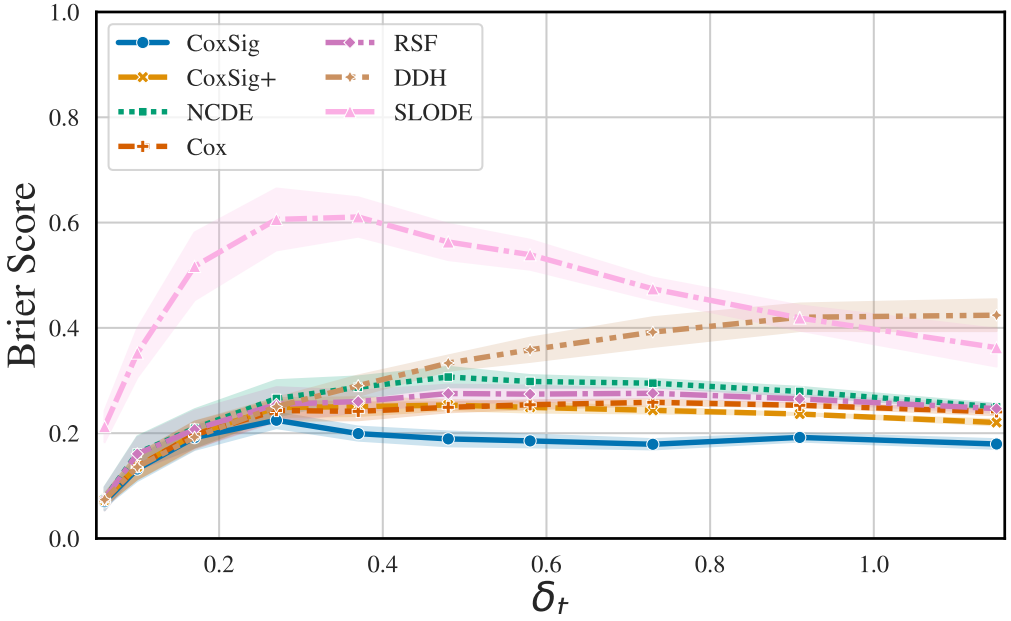

Figure 4.4 – Brier score $\delta t \mapsto \mathrm{BS}(t, \delta t)$, evaluated at $t = 0.23$, for the partially observed SDE experiment. Confidence intervals indicate 1 standard deviation.

Tumor Growth. We similarly aim at predicting the hitting time of a stochastic process modelling the growth of a tumor (Simeoni et al., 2004), where x^i represents a drug-intake. In this experiment, the time series $\mathbf{X}^i$ is very-low dimensional ($d = 2$, which includes the time channel).

4.4.5 Real-World Datasets

Predictive Maintenance. (Saxena et al., 2008) This dataset collects simulations of measurements of sensors placed on aircraft gas turbine engines run until a threshold value is reached. In this context, the time-to-event is the failure time. This dataset features a small sample size, considerable censoring rates and a high number of time channels.

Churn prediction. We use a private dataset provided by Califrais, a food supply chain company that delivers fresh products from Rungis to food professionals. The company has access to a variety of features observed through time for every customer. Its goal is for example to predict when the customer will churn. The time series in this setup are high dimensional but sampled at a low frequency.

	Algorithms	Avg. C-Index ↑	IBS ↓
OU	CoxSig	**0.857±0.01**	**0.091±0.01**
	CoxSig+	0.857±0.01	0.095±0.01
	NCDE	0.517±0.04	0.103±0.01
	DDH	0.545±0.02	0.094±0.01
	SLODE	0.621±0.05	0.253±0.03
Tumor	CoxSig	0.696±0.02	0.138±0.01
	CoxSig+	0.797±0.03	0.137±0.01
	NCDE	0.827±0.02	**0.130±0.01**
	DDH	**0.941±0.05**	0.133±0.01
	SLODE	0.601±0.07	0.136±0.01
NASA	CoxSig	0.858±0.04	0.154±0.03
	CoxSig+	**0.867±0.04**	0.154±0.03
	NCDE	0.541±0.09	0.178±0.04
	DDH	0.813±0.06	0.156±0.02
	SLODE	0.438±0.14	**0.145±0.02**
Califrais	CoxSig	0.741±0.01	0.130±0.01
	CoxSig+	**0.751±0.01**	**0.129±0.01**
	NCDE	0.529±0.05	0.152±0.01
	DDH	0.570±0.03	0.139±0.01
	SLODE	0.542±0.03	0.193±0.03

Table 4.2 – Averaged value of our metrics for 4 considered dataset over set of 10 different values of t chosen from the 5 to the 50th percentile of the distribution of event times. The values of δt for each dataset is chosen to be the same as that shown in Figure 4.5.

Further details on all datasets are provided in Appendix B.4. Overall, our datasets are diverse in terms of sample size, size and length of the time series and censoring type.

4.4.6 Results

General performance of CoxSig. Overall, the signature-based estimators outperform competing methods. We observe that CoxSig and CoxSig+ improve over the strongest baselines in terms of Brier scores. Contrarily to the strong baseline DDH, this improvement is consistent over larger prediction windows $[t, t + \delta t]$ as δt increases (see Figure 4.4). They provide even stronger improvements in terms of C-indexes (see Figure 4.5 and Appendix B.4). This suggests that they are particularly well-tailored for ranking tasks, such as identifying the most-at-risk individual. Including the first observed value of the time series generally improves CoxSig's performance: this is possibly due to the fact that signatures are invariant by translation (i.e. the signature of $x : t \mapsto x(t)$ is equal to the signature of $x : t \mapsto x(t) + a$), and hence including the first value of the time series provides non-redundant information.

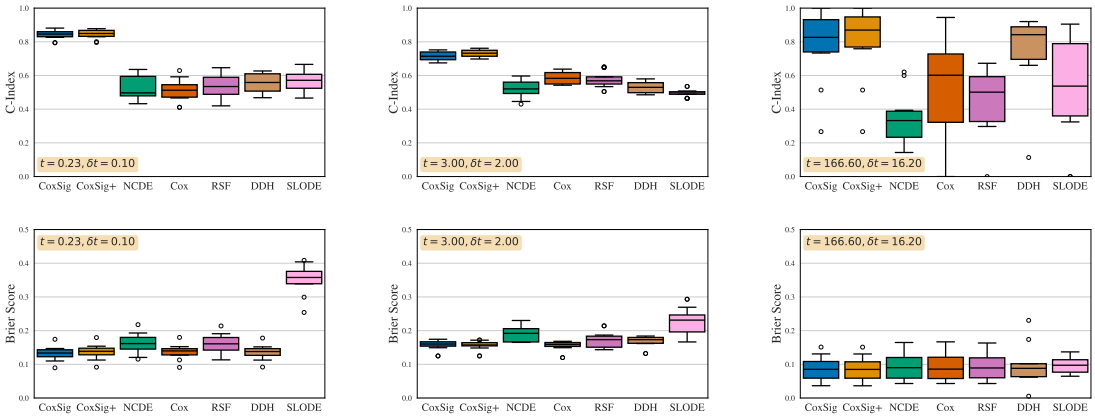

Figure 4.5 – C-Index (*higher* is better) on **top** and Brier score (*lower* is better) on **bottom** for hitting time of a partially observed SDE (**left**), churn prediction (**center**) and predictive maintenance (**right**) evaluated at chosen points $(t, \delta t)$. t is chosen as the first decile of the event times i.e. 90% of the events occur after t. Hollow dots indicate outliers, and error bars indicate 80% of the interquartile range. We report detailed results for numerous points $(t, \delta t)$ in Appendix B.4.

Performance on low-dimensional data. A notable exception is the tumor growth simulation, in which CoxSig is generally outperformed (see Figures B.10 and B.11 in the appendix). The competitive performance of signatures for moderate to high dimensional data streams and its below average performance on low dimensional data is a well-studied feature (see Fermanian (2021) for an empirical study). A possible solution to handle low-dimensional data is to use embeddings before computing signatures to make them more informative (Morrill et al., 2020).

NCDEs. On the other side, NCDEs generally tie or perform worse than competing methods. Notably, when considering C-indexes, they even perform worse than random on the predictive maintenance dataset. This stands in stark contrast to their good performances on classification or regression tasks (Kidger et al., 2020; Morrill et al., 2021; Vanderschueren et al., 2023).

Running times. Finally, we observe that our methods run in similar times than DDH, while including cross-validation (see Figure B.8 in the appendix). Models that do not use time dependent features (RSF and Cox) are 2 orders of magnitude faster to train.

4.5 Conclusion

We have designed and analyzed a model for generic counting processes driven by a controlled latent state, which can be readily estimated using either NCDE or signature-based estimators. CoxSig in particular offers a parsimonious alternative to deep models and

yields excellent performance for survival analysis. Future research efforts will be targeted at extending our model to competing risks and multimodal data.

Limitations. While our model shows competitive performance on moderate to high-dimensional data, one central limitation is its below average performance on low dimensional data. We also stress that the extension to very high dimensional time series is computationally prohibitive since the signature scales exponentially with the dimension of the time series. Finally, our experimental setup is limited to the survival analysis case: we plan on extending it to general counting processes in future work.

Chapter 5

Comparison of classification and survival models for dynamic churn prediction

Contents

Abstract In this paper, we address the churn prediction problem by leveraging both demographic and customer behavior data. Our approach involves training models across

three different approaches. First, we frame the churn prediction task as a binary classification problem to directly identify whether a customer is likely to churn within a given time frame. Second, we explore various survival analysis frameworks, which not only predict whether a customer will churn but also estimate the time until churn. Third, we extend the application of survival analysis by incorporating the landmark setting, which simplifies the training process and makes it easier to scale for high-dimensional datasets. For each of these approaches, we implement a range of models and advanced feature engineering techniques to optimize predictive performance. We conduct extensive experiments using historical order data from the company Califrais, along with a thorough comparison of these approaches. Our evaluation highlights the strengths and weaknesses of each approach in the context of real-time churn prediction, offering valuable insights for both academic research and practical applications.

5.1 Introduction

5.1.1 Churn prediction

Customer retention is a critical factor in the success and sustainability of any business, particularly in industries where recurring revenue from existing customers is a significant component of profitability. One of the major challenges companies face is customer churn, which occurs when customers stop using a product or service, either by canceling a subscription, switching to a competitor, or simply disengaging. According to the study conducted by Mozer et al. (2000), marketing campaigns for retaining existing customers provide a better return on investment than putting efforts into attracting new customers. Therefore, churn prediction, which helps to identify in advance those customers who might become churners, together with the potential reasons for their dissatisfaction, is essential for businesses aiming to maintain a stable customer base and optimize their marketing and customer service efforts. Client churn studies have been conducted across various service sectors, for example, games (Periáñez et al., 2016), telecommunication (Gui, 2017), or finance (Larivière and Van den Poel, 2004).

5.1.2 Overview of churn prediction algorithms

This churn prediction task poses a challenge in management science that can be tackled either manually, through human analysis, or automatically, by leveraging advanced techniques such as machine learning. For automated approaches, data is fundamental, and it can come from different sources, such as demographic information or customer behavior.

Learning with customer's behavior data. Demographic data, which includes elements such as age, gender, income, and location, can be valuable for segmenting customers into different levels of churn risk. In addition, this data is straightforward to analyze and interpret, making it easier to understand and apply in creating targeted retention strategies (Mittal and Kamakura, 2001; Athanassopoulos, 2000). However, access to this data is not always available—either because it was not collected, customers chose not to share it, or due to restrictive privacy laws. In contrast, customer behavior data (e.g. the number of orders, the number of items per order, the total amount spent by the customer, etc) is consistently produced and easily accessible during business operations, making it a more practical and reliable source for developing predictive models. Moreover, customer behavior data, with its dynamic nature, continuously updates to reflect real-time interactions and changes in customer engagement, making it more effective for predicting churn than static demographic data. This consistent and real-time availability of behavioral data provides a robust foundation for developing accurate and dynamic churn predictions, allowing the business to respond effectively and timely to evolving customer behaviors (Buckinx and Van den Poel, 2005; Burez and Van den Poel, 2007). However, integrating this type of data into a machine learning model poses significant challenges due to its dynamic and complex nature, including correlated observations and non-linear trajectories that evolve over time.

Effectively capturing these complicated temporal patterns requires applying advanced techniques designed to handle such complexity and ensure the accuracy and robustness of the model.

From binary approach to temporal approach. Many of the previous studies have approached churn prediction as a binary classification problem, predicting whether a client will churn within a specified time frame, see e.g. Buckinx and Van den Poel (2005), Coussement and Van den Poel (2008), Verbeke et al. (2012), and Zhang et al. (2017). This approach consists of labeling each client as a churner or not, allowing the problem to fall into the large domain of supervised learning, with many algorithms available. However, this labeling process can vary depending on the specific business context.

In contractual settings, churn typically refers to clients who do not renew their contracts when they expire, leading to predictions about whether a client will churn at the renewal date based on their historical activity, see e.g. Coussement and Van den Poel (2008) and Zhang et al. (2017).

In non-contractual settings, where no formal contract binds the client to the company, clients can leave at any time without restrictions. In such cases, a specific churn criterion is constructed. For instance, if a client stops using a service for a certain period, known as the churn window, they are considered as a churn client. The churn prediction then turns to the classic binary classification as in the contractual case above, see e.g. Buckinx and Van den Poel (2005) and Verbeke et al. (2012).

Although this method simplifies churn prediction, its performance in non-contractual contexts heavily depends on the choice of the churn criterion. In addition, in many services, estimating the survival time, i.e., the time elapsed before the client churns, is critical for timely interventions and efficient resource allocation. However, this binary classification approach fails to distinguish between clients who churn early in the churn window and those who churn later, nor can it predict the survival time of clients who have not yet churned (Khodadadi et al., 2020).

To overcome the limitations of the classification approach, alternative methods have been proposed for predicting the time until a client churns. When the survival times of all clients are observed, standard regression models for continuous outcomes can be used to predict churn timing (Buis, 2006). However, in many instances, either due to limited historical data or ongoing client engagement, churn events are not fully observed, resulting in censored data. Hence, it is essential to develop models that not only estimate churn timing but also effectively handle the issue of censoring. Consequently, churn prediction falls within the scope of survival analysis, and it is of interest to explore its performance in comparison to the classification approach.

5.1.3 Contributions

To develop efficient models for churn prediction based on customer behavior data, we propose a comprehensive approach by training models across three approaches. First,

we frame the churn prediction task as a binary classification problem, to directly identify whether a customer is likely to churn within a given time frame. Second, we explore various survival analysis frameworks, which not only predict whether a customer will churn but also estimate the time until churn. Third, we extend the application of survival analysis by incorporating the landmark setting, which simplifies the training process and makes it easier to scale for high-dimensional datasets. For each of these approaches, we implement a range of models and advanced feature engineering techniques to optimize predictive performance. Finally, we conduct a comprehensive comparison of these approaches, evaluating their strengths and weaknesses in the context of real-time churn prediction. Overall, the CoxSig model demonstrates particularly strong predictive performance. This comparative analysis across the three approaches, which has not been previously conducted in the literature, provides valuable insights into the efficacy of different approaches, offering guidance for both academic research and practical application in churn management.

Outline. Section 5.2 presents an overview of churn prediction at Califrais, establishing the foundation for notation and modeling setup. In Section 5.3, we describe the specific framework that will be implemented for churn prediction. Lastly, Section 5.4 is dedicated to a detailed evaluation of the prediction performance, assessing the effectiveness and accuracy of the proposed models.

5.2 Context and mathematical setting

5.2.1 Business context

Califrais company. Founded in July 2014, Califrais aimed to modernize the distribution of fresh food from the Rungis market to restaurants in Paris by addressing inefficiencies in traditional supply chains. These inefficiencies, such as manual processes and opaque pricing, resulted in wasted time, food, and CO2 emissions, particularly in the perishable fresh food sector. Califrais' solution mutualizes orders from multiple suppliers into one delivery, optimizing both customer and supplier operations. By 2021, Califrais launched the official digital marketplace for Rungis, *https://rungismarket.com*, providing customers with detailed product information and simplifying the ordering process. Over a decade, the company has grown its product catalog to over 8,000 items across 120 categories, expanding its reach beyond Paris and France.

Business context. At Califrais, the client orders are recorded over time. Figure 5.1 shows the historical orders of 3 clients within an observed period of 2 years, which are grouped by week.

In this figure, we observe that while the clients can start ordering at any time, some clients keep ordering frequently (client A), some clients stop ordering for a long time (client C in August 2022) and then re-order or even do not order anymore from a specific point (client B after April 2022). At Califrais, our goal is to prevent these long periods of inactivity,

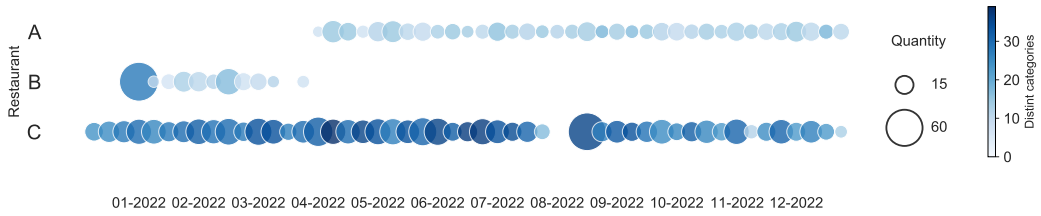

Figure 5.1 – Order history of 3 clients (A, B, and C), acquired in 2022, differing in the quantity of products ordered, and by the number of categories in which these products were ordered.

as seen with clients B and C, by identifying these clients early and re-engaging them with our services promptly. Moreover, these different ordering behaviors play a crucial role in shaping our definition of churn.

Historical order data. At Califrais, beyond classical static data related to the customer such as location or business size, it has been observed that the historical order data is strongly related to potential customer churn. This type of data, which is tracked over time—known as longitudinal data—also shows a similar relationship with the risk of customer churn, as discussed in studies by Wei and Chiu (2002), Buckinx and Van den Poel (2005), and Alboukaey et al. (2020).

Figure 5.2 shows an illustration of how these features can affect the risk of churn. A detailed description of the dataset is given in Section 5.4.1.

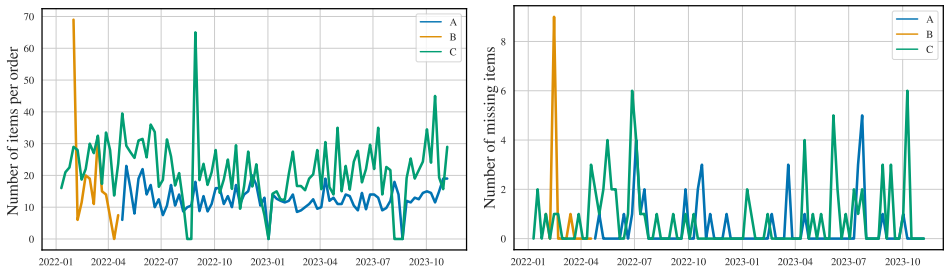

Figure 5.2 – The historical information of the number of items and the number of missing items of each order for 3 clients A, B, and C in Figure 5.1. The high number of missing items on the first orders of client B could be the reason they churn quickly after several orders while the low number of missing items of clients A and B makes them stay longer.

Churn definition. We define a client as having churned if they have not placed any orders for 4 consecutive weeks. Within this definition, a phenomenon known as recurring churn can occur, where a client repeatedly churns and then returns to the company. To simplify churn prediction modeling, we treat clients who order again after churning as new clients. Within the observed period, clients who do not churn are considered censored.

Churn prediction in real-time context. Based on the above churn definition, we now have a dataset comprising both clients who have already churned and those who are still actively engaged, along with their historical order data. The goal is to develop a model capable of accurately predicting, at any time, the probability that currently engaged clients are likely to churn. In addition, this model needs to be robust and adaptable, ensuring it can effectively predict the churn probability for new clients from the moment they join and throughout their engagement period.

5.2.2 Mathematical context

Dataset notation. In all the following, we consider a set of n clients in our dataset $\mathcal{D}^n$. For each client $i \in 1, \dots, n$, based on historical order data from Califrais, we define a set of q static features denoted by $Z^i \in \mathbb{R}^q$. In addition, order-related features are grouped by week, with the first week the client i places an order represented by s_1^i. We denote by $T^i \in \mathbb{N}^*$ the number of weeks from s_1^i until the client either churns or is censored. The entire order history for client i, which includes d longitudinal features measured from the first ordering week s_1^i over T^i weeks, is denoted by the set $\{X^i(s_1^i), \dots, X^i(s_1^i + T^i)\} \in \mathbb{R}^{d \times T^i}$. Moreover, the survival indicator Δ^i equals 1 if this client has churned at its last observed week $s_1^i + T^i$ or not, in other terms

$$\Delta^i = \mathbb{1}_{s_1^i + T^i < s_\tau},$$

where s_τ is the end of the period where the data are collected for the analysis. We gather all these data and define

$$\mathcal{D}^n = \left\{ \left(s_1^i, X^i(s_1^i)\right), \dots, \left(s_1^i + T^i, X^i(s_1^i + T^i)\right), Z^i, T^i, \Delta^i \right\}_{i=1}^n.$$

Risk prediction. In a real-time context, let s_{P} represent the prediction week, and consider an active client i who has continued ordering up to s_{P}. The objective at this point is to predict the probability that this client will churn within the next δ_t weeks, based on their historical data up to week s_{P}, denoted as $X^i(s_1^i), \dots, X^i(s_{\text{P}})$. We represent this probability as $\mathcal{R}^i(s_{\text{P}}, \delta_t)$.

Train-Test data. To ensure that the learning model performs effectively in a real-time context-where it is trained on all historical data and used to predict future events-we divide the dataset in a similar way, using a specific time point to separate it into a training set for model training and a test set for evaluating prediction performance. Let us denote by $|A|$ the cardinality of a set A. The train set $\mathcal{D}^n_{\text{train}}$ and the test set $\mathcal{D}^n_{\text{test}}$, which are split from $\mathcal{D}^n$, and the time point at which the split is done s_κ , are then respectively defined as

$$\mathcal{D}^n_{\text{train}} = \left\{ \left(s_1^i, X^i(s_1^i)\right), \dots, \left(s_1^i + T^i_{\text{train}}, X^i(s_1^i + T^i_{\text{train}})\right), Z^i, T^i_{\text{train}}, \Delta^i_{\text{train}} \mid s_1^i \leq s_\kappa \right\}_{i=1}^n,$$

where

$$T^i_{\text{train}} = |\{s_1^i, \dots, \min(s_1^i + T^i, s_\kappa)\}| \quad \text{and} \quad \Delta^i_{\text{train}} = \mathbb{1}_{s_1^i + T^i \leq s_\kappa},$$

and

$$\mathcal{D}^n_{\text{test}} = \left\{ \left(s^i_1, X^i(s^i_1)\right), \ldots, \left(s^i_1 + T^i, X^i(s^i_1 + T^i)\right), Z^i, T^i, \Delta^i \mid s^i_1 + T^i > s_\kappa \right\}^n_{i=1}.$$

It is important to note that the testing set includes two types of clients: those who are in the training set but continue ordering after s_κ and those who begin ordering only after s_κ.

From calendar time to client time. To simplify modeling, the dataset in calendar time should be converted to client time, meaning we focus on the elapsed time or duration since the starting date. At week $s > s^i_1$ in calendar time, the equivalent time point t^i for a client i in client time can be defined as

$$t^i = |\{s^i_1, \ldots, s\}|.$$

We show an illustration in Figure 5.3 below.

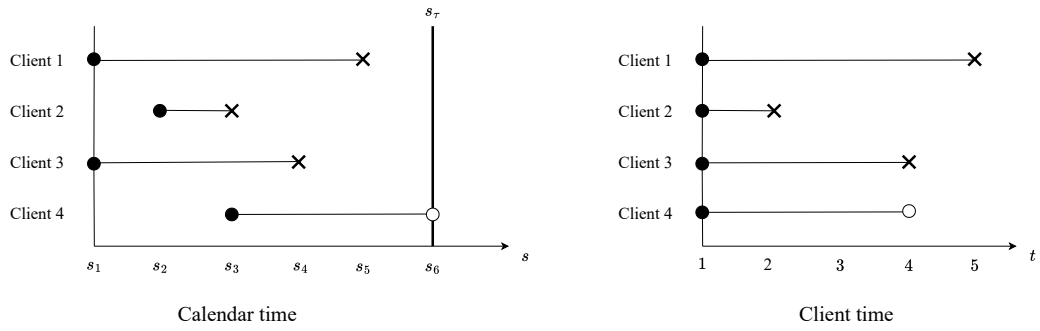

Figure 5.3 – Convert the data from calendar time (**left**) to client time (**right**). The black dot represents the first week a client orders, the cross represents the week it churns and the white represents the censored client at the last observed week.

The historical order $\{X^i(s^i_1), \ldots, X^i(s^i_1 + T^i)\}$ of client i in the calendar time can be redefined in the client time as $\{X^i_1, \ldots, X^i_{T^i}\}$. The dataset $\mathcal{D}^n$ is then rewritten as $\mathcal{D}^n_{\text{surv}}$, which is

$$\mathcal{D}^n_{\text{surv}} = \left\{ X^i_1, \ldots, X^i_{T^i}, Z^i, T^i, \Delta^i \right\}^n_{i=1}.$$

Note that if $\Delta^i = 1$, then T_i corresponds to the churn time.

With a slight abuse of notation, we denote by $\mathcal{R}^i(t^i_{\text{P}}, \delta_t)$ the risk that the client i churns in the next δ_t weeks if they have not churned yet after t^i_{P} weeks in client time, where $t^i_{\text{P}} = |\{s^i_1, \ldots, s_{\text{P}}\}|$.

5.3 Methods

To develop efficient models for churn prediction in the real-time context, we implement models across three approaches: the binary approach, the temporal approach, and

the landmarking approach. For each of these approaches, we implement a range of models and advanced feature engineering techniques to optimize predictive performance. Details of these methods are discussed in the following sections.

5.3.1 Binary approach

We begin by seeing the problem as a supervised learning task with binary labels. The goal is then to predict whether a client will churn at a specific time t. It requires defining covariates and labels for training the model. Starting from the original dataset $\mathcal{D}^n_{\text{surv}}$, we generate a new dataset suitable for applying classification machine learning techniques. The process for generating this new dataset is described below.

Churn label definition. For each client $i \in \{1, \dots, n\}$, we denote $L^i(t, \delta_t)$ as the churn label, which defines whether client i churns in the period $[t,\ t + \delta_t]$. We then have

$$L^i(t, \delta_t) = \mathbb{1}_{t<T^i\leq t+\delta_t, \Delta^i=1}.$$

Figure 5.4 shows an example of how to define the churn label.

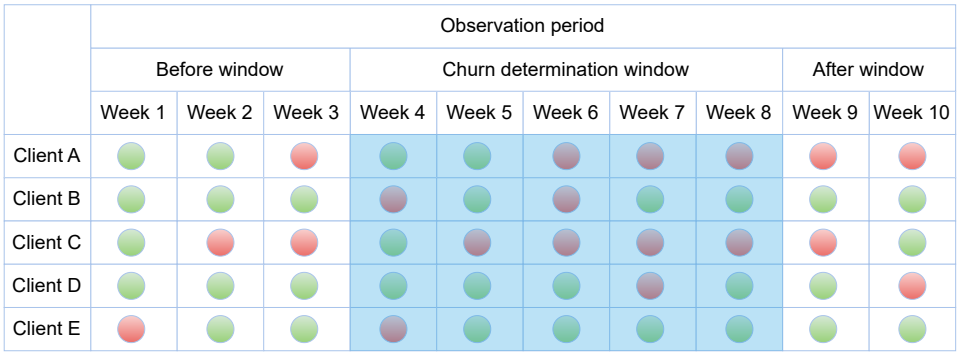

Figure 5.4 – Schematic of the time window method used for churn prediction in the binary approach. The data represents 5 clients over a 10-week period, where a red dot indicates no order was placed in a given week and a green dot signifies an order was placed. According to the churn definition, which considers a client to have churned if no orders are placed for 4 consecutive weeks, clients A and C are identified as having churned at weeks 6 and 5, respectively, while clients B, D, and E are censored at week 10. To illustrate the process of defining the churn label in the binary approach, the time window of length $\delta_t = 5$, which is in the blue area, is set from week 4 $(t = 4)$ to week 8 $(t + \delta_t = 8)$. Since clients A and C churn within this window, they are labeled as churned, while clients B, D, and E are labeled as non-churned.

Note that the time range for defining the churn label varies depending on whether the client is censored. For censored clients, t is chosen from the range $\{1, \dots, T^i - \delta_t\}$, whereas for non-censored clients, it is selected from $\{1, \dots, T^i\}$.

Feature engineering. Based on the historical longitudinal information of client i up to t, denoted by $X^i_{1:t} = \{X^i_1, \dots, X^i_t\}$, we apply a set of $M \in \mathbb{N}^+$ known feature engineering functions $\Psi_m : X^i_{1:t} \to \Psi_m(X^i_{1:t}) \in \mathbb{R}^{d'}$, $m \in \{1, \dots, M\}$ to extract a relevant set of covariates, where d' is the dimension of extracted features and will vary depending on the type of feature engineering functions. In this work, we take various forms of the feature engineering function. For example, it might use the value of the longitudinal feature at week t:

$$\Psi_m(X^i_{1:t}) = X^i_t,$$

or the value of the longitudinal feature one week before t:

$$\Psi_m(X^i_{1:t}) = X^i_{t-1},$$

or the duration from the client's first order up to week t, known as the level of loyalty:

$$\Psi_m(X^i_{1:t}) = t.$$

Alternatively, we may apply a truncated signature transformation $\mathbf{S}_N$ of order $N \geq 1$, as defined, for example, in Kidger et al. (2019, Definition 1.1) and introduced in Section 2.2.4:

$$\Psi_m(X^i_{1:t}) = \mathbf{S}_N(X^i_{1:t}).$$

In addition, some packages, such as `tsfresh`, offer to extract a comprehensive set of feature engineering functions Ψ_m of longitudinal features. Simple examples of such functions are the maximum or the sum of the longitudinal features, respectively defined by

$$\Psi_m(X^i_{1:t}) = \max\left\{X^i_1, \dots, X^i_t\right\}, \quad \Psi_m(X^i_{1:t}) = \sum_{u=1}^{t} X^i_u.$$

Dataset construction. Let us define by $\tilde{X}^i_t = \big(\Psi_1(X^i_{1:t}), \dots, \Psi_M(X^i_{1:t})\big)$ the set of covariates extracted from the order history of an active client i up to time t, using a set of specific feature engineering functions $\{\Psi_1, \dots, \Psi_M\}$. The new dataset , denoted as $\mathcal{D}^n_{\text{bin}}$, is the combination of the set of covariates $\tilde{X}^i_t$, and the churn label $L^i(t, \delta_t)$ for different time $t \in \{1, \dots, T^i\}$ and all client $i \in \{1, \dots, n\}$. This new dataset is then defined as

$$\mathcal{D}^n_{\text{bin}} = \left\{\big((\tilde{X}^i_1, Z^i), L^i(1, \delta_t)\big), \dots, \big((\tilde{X}^i_{T^i}, Z^i), L^i(T^i, \delta_t)\big)\right\}_{i=1}^{n}.$$

Learning classifier. In this work, we apply two standard classifiers, which are the *logistic regression* and the *random forest* on the dataset $\mathcal{D}^n_{\text{bin}}$. For a given classifier, let F denote the decision function, parameterized by a learnable parameter θ, which maps the input feature vector $(\tilde{X}^i_t, Z^i)$ to a label prediction. The risk prediction $\mathcal{R}^i(t^i_{\text{p}}, \delta_t)$ that client i churns in the next δ_t weeks if they have not churned after t^i_{p} weeks, is then defined as

$$\mathcal{R}^i(t^i_{\text{p}}, \delta_t) = F_\theta\big((\tilde{X}^i_{t^i_{\text{p}}}, Z^i)\big).$$

5.3.2 Temporal approach

As previously noted, the disadvantage of the binary approach is its inability to estimate the time until a client churns, known as churn time. Survival analysis offers an alternative approach by modeling this churn time while addressing the issue of censoring. In this section, we present several survival models that have been employed in this work.

General framework. In this framework, the churn time of client $i \in \{1, \dots, n\}$ is assumed to be a non-negative random variable and denoted by $\tilde{T}^i$. The relation between the survival function S^i and the hazard rate λ^i is defined by

$$S^i(t) = \mathbb{P}(\tilde{T}^i \geq t) = \exp\big(-\int_0^t \lambda^i(s)ds\big). \tag{5.1}$$

The objective is to build a parameterized model $\{\lambda^i_\theta; \theta \in \Theta\}$ that learns the distribution of churn time through the hazard function λ^i, governed by the parameter θ. The best parameter in our model can be estimated by maximizing the log-likelihood function on the dataset $\mathcal{D}^n_{\text{surv}}$, which is defined as

$$\mathcal{L}(\theta \,|\, \mathcal{D}^n_{\text{surv}}) = \sum_{i=1}^{n} \Delta^i \log \lambda^i_\theta(T^i) - \int_1^{T^i} \lambda^i_\theta(u)du.$$

In this work, we model the parameterized hazard function λ^i_θ for individual i influenced by both static features Z^i and longitudinal data X^i, which is $\lambda^i_\theta(t) = \lambda_\theta(t|X^i_t, Z^i)$.

Risk prediction. Given the estimated distribution of churn time, the risk prediction $\mathcal{R}^i(t^i_{\text{P}}, \delta_t)$ that client i churns in the next δ_t weeks if they have not churned after t^i_{P} weeks, is then defined as

$$\begin{aligned}
\mathcal{R}^i(t^i_{\text{P}}, \delta_t) &= \mathbb{P}(t^i_{\text{P}} < \tilde{T}^i < t^i_{\text{P}} + \delta t \,|\, \tilde{T}^i > t^i_{\text{P}}, X^i_{1:t^i_{\text{P}}}, Z^i) \\
&= \frac{\mathbb{P}(t^i_{\text{P}} < \tilde{T}^i < t^i_{\text{P}} + \delta t \,|\, X^i_{1:t^i_{\text{P}}}, Z^i)}{\mathbb{P}(\tilde{T}^i > t^i_{\text{P}} \,|\, X^i_{1:t^i_{\text{P}}}, Z^i)} \\
&= \frac{S_\theta(t^i_{\text{P}} \,|\, X^i_{1:t^i_{\text{P}}}, Z^i) - S_\theta(t^i_{\text{P}} + \delta t \,|\, X^i_{1:t^i_{\text{P}}}, Z^i)}{S_\theta(t^i_{\text{P}} \,|\, X^i_{1:t^i_{\text{P}}}, Z^i)},
\end{aligned}$$

where the above survival function S_θ can be derived from (5.1) by

$$S_\theta(t^i_{\text{P}} + \delta t \,|\, X^i_{1:t^i_{\text{P}}}, Z^i) = \exp\Big(-\int_1^{t^i_{\text{P}}+\delta t} \lambda_\theta(u \,|\, X^i_{1:(u \wedge t^i_{\text{P}})}, Z^i)du\Big).$$

In the following, we provide a description of the specific survival analysis frameworks employed in this work.

CoxSig. In this framework, the hazard function is assumed to take the Cox model form (Cox, 1972b) and depends on both time-independent features and time-dependent features. In addition, the time-dependent features are extracted by applying the truncated signature transform $\mathbf{S}_N$ on longitudinal data. The hazard function is then in the form

$$\lambda_\theta(t|X^i_{1:t}, Z^i) = \exp\big(\mathbf{S}_N(X^{i,\text{sig}}_{1:t})^\top \alpha + {Z^i}^\top \beta\big),$$

where $\theta = (\alpha, \beta)$, and $X^{i,\text{sig}}_{1:t} = \big\{\big(1, X^i_1\big), \ldots, \big(t, X^i_t\big)\big\}$ is the extension of $X^i_{1:t}$ by adding the time dimension. Given the dataset $\mathcal{D}^n_{\text{surv}}$, the log-likelihood of the model can be defined as

$$\mathcal{L}(\theta \,|\, \mathcal{D}^n_{\text{surv}}) = \sum_{i=1}^{n} \Big(\Delta^i \log \lambda_\theta(T^i|X^i_{1:T^i}, Z^i) - \int_1^{T^i} \lambda_\theta(u|X^i_{1:u}, Z^i)du\Big).$$

Denoting by $\hat{\theta}$ a maximizer of this log-likelihood, we can form the predictions from the estimated survival function

$$S_{\hat{\theta}}(t + \delta t|X^i_{1:t}, Z^i) = \exp\Big(-\int_1^{t+\delta t} \lambda_{\hat{\theta}}(u|X^i_{1:(u\wedge t)}, Z^i)du\Big).$$

We refer the reader to Chapter 4 for a more detailed description.

FLASH - joint models. Joint models is a powerful statistical tool widely used in biostatistics and medical research (Wulfsohn and Tsiatis, 1997; Proust-Lima et al., 2009; Andrinopoulou and Rizopoulos, 2016) simultaneously analyze longitudinal data and survival data, allowing to understand how the progression of a longitudinal marker influences the risk of an event occurring. This is achieved through the integration of a longitudinal submodel and a survival submodel, which are linked via a common latent structure. Building on the strengths of this framework, we propose applying an advanced joint model, which is called FLASH, to tackle the challenge of the churn prediction. We summarize here the main concepts of FLASH framework; for a more detailed description, we refer the reader to Chapter 3 of this manuscript or Nguyen et al. (2023).

In the FLASH framework, the three submodels are in the form: a multinomial logistic regression defining the probability of belonging to a latent class, a generalized linear mixed model for each latent class describing the evolution of the longitudinal markers, and finally a Cox class-specific survival model.

The population of n clients is assumed to be heterogeneous, consisting of $K \in \mathbb{N}^*$ latent classes representing the different risk levels of a client. To each subject $i \in \{1, \ldots, n\}$, we associate a categorical latent variable $g_i \in \{1, \ldots, K\}$, which encodes its latent class membership. Then, the latent class membership probability is assumed to take the form,

for any $k \in \{1, \dots, K\}$,

$$\mathbb{P}(g^i = k) = \frac{e^{Z^{i\top}\xi^k}}{\sum_{j=1}^{K} e^{Z^{i\top}\xi^j}},$$

where $\xi^k \in \mathbb{R}^p$ denotes a vector of coefficients for class k. Each latent class is characterized by a class-specific longitudinal model and a class-specific survival model, which are described in the following.

The class-specific longitudinal model is described by a standard linear mixed model (Laird and Ware, 1982)

$$X^i(t)|g^i = k \sim \mathcal{N}\big(U(t)\beta^k + V(t)b^i, \Sigma\big)$$

where Σ is a variance-covariance matrix of measurement errors, $U(t) \in \mathbb{R}^{d\times q}$ is a matrix of time-varying features with corresponding unknown fixed effect parameters $\beta^k \in \mathbb{R}^q$, and $V(t) \in \mathbb{R}^{d\times r}$ is a matrix of time-varying features with corresponding random effect $b^i \sim \mathcal{N}(0, D)$, with $D \in \mathbb{R}^{r\times r}$ being a the variance-covariance matrix of random effects.

To quantify the effect of the longitudinal data on the churn time, this longitudinal data is represented by a set of $M \in \mathbb{N}^+$ known functionals $\{\Psi_1, \dots, \Psi_M\}$ similar to the one defined in Section 5.3.1. The class-specific survival model, which is in the form of the Cox model (Cox, 1972b), is then defined as

$$\lambda(t \,|\, X^i_{1:t^-}, g^i = k) = \lambda_0(t) \exp\big(\tilde{X}^i_{1:t^-}\gamma_k\big),$$

where λ_0 is an unspecified baseline hazard function that does not depend on k and γ_k the joint representation parameters, which are the only class-specific objects in this model.

From all the submodels described above, for a fixed number of latent classes K, the log-likelihood $\mathcal{L}(\theta)$ on the dataset $\mathcal{D}^n_{\text{surv}}$ can be decomposed using the conditional independence assumptions, which can be defined as

$$\mathcal{L}(\theta \,|\, \mathcal{D}^n_{\text{surv}}) = \sum_{i=1}^{n} \log\Big(\sum_{k=1}^{K} \mathbb{P}_\theta(g^i = k) f_\theta(T^i|g^i = k) \lambda_\theta(T^i|X^i, g^i = k)^{\Delta^i} S_\theta(T^i|g^i = k)\Big),$$

where θ is the combination of all unknown parameters in the model, $S_\theta(T^i|g^i = k)$ is the corresponding class-specific survival function derived from $\lambda_\theta(T^i|X^i, g^i = k)$ and $f_\theta(T^i|g^i = k)$ is the density of the longitudinal data in class k. We refer the reader to Chapter 3 for a more detailed description.

Dynamic DeepHit (Lee et al., 2019). DDH is a state-of-the-art method for dynamical survival analysis, that combines an RNN with an attention mechanism and uses both time-dependent and static features.

Random Survival Forest (RSF). Proposed by Ishwaran et al. (2008), RSF is an ensemble learning method for survival analysis that extends the concept of random forests (RF) (Breiman, 2001) to handle time-to-event data. The implementation of RSF (Ishwaran et al.,

2008) only considers static features as the input. To incorporate longitudinal features X^i alongside the static features Z^i, we use feature engineering techniques detailed in Section 5.3.1 to effectively aggregate the longitudinal features into static features, enabling their integration into the RSF framework.

5.3.3 Landmark training

To obtain the risk prediction for clients who have not churned after a fixed number of weeks t_{P} in the survival analysis framework, the model can be trained in a landmark fashion (Anderson et al., 1983; Van Houwelingen, 2007; Devaux et al., 2022) where we use a set of only clients whose churn time is greater than the time t_{P} and discard the longitudinal data after t_{P} (see Figure 5.5 for an illustration). In this approach, we call this fixed number of weeks t_{P} as the landmark time. This landmarking approach is computationally simple and scales effectively for high-dimensional problems; however, it inefficiently uses data and requires retraining with updated information at each new prediction time. Furthermore, in real-time prediction contexts, where clients may have varying elapsed times up to a given prediction week, multiple models need to be trained at respective landmark times, adding complexity and increasing the computational cost.

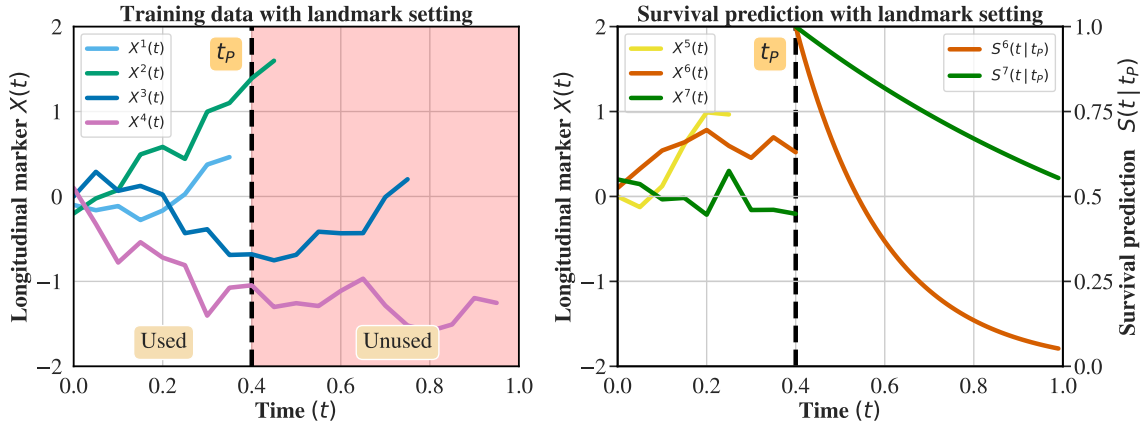

Figure 5.5 – Learning in landmark setting with longitudinal data $X(t)$ of 7 clients. We set the landmark time t_{P} to 0.4. On the **left**, we select clients whose churn times are greater than the landmark time, which are the clients $i \in \{2, 3, 4\}$ and we ignore the client $i = 1$. The longitudinal markers observed only up to the landmark time are used to train the survival model while we ignore the remaining which past the landmark time. On the **right**, we predict the conditional survival probability $S^i(t|t_{\mathrm{P}}) = \mathbb{P}(\tilde{T}^i > t \,|\, \tilde{T}^i > t_{\mathrm{P}}, X^i_{1:t_{\mathrm{P}}}, Z^i)$ of clients i who survive at landmark time with their longitudinal markers observed up to landmark time. In this case, we ignore the client $i = 5$ and derive the simulated survival probability prediction of client $i \in \{6, 7\}$.

CoxSig with landmark training. At the landmark time t_{P}, the hazard function is in form

$$\lambda_\theta(t|X^i_{1:t_{\mathrm{P}}}, Z^i) = \begin{cases} \exp\left(\mathbf{S}_N(X^{i,\mathrm{sig}}_{1:t})^\top \alpha + {Z^i}^\top \beta\right) & \text{if } t_{\mathrm{P}} > t \\ \exp\left(\mathbf{S}_N\{\mathrm{LOCF}(X^{i,\mathrm{sig}}_{1:t_{\mathrm{P}}}, t)\}^\top \alpha + {Z^i}^\top \beta\right) & \text{otherwise} \end{cases}$$

where $\theta = (\alpha, \beta)$, and $\mathrm{LOCF}(X^{i,\mathrm{sig}}_{1:t_{\mathrm{P}}}, t)$ is the last observation carried forward (LOCF) version of $X^{i,\mathrm{sig}}_{1:t_{\mathrm{P}}}$. It is constructed by filling forward the last observed value from t_{P} up to t while letting the time dimension evolve, which is

$$\mathrm{LOCF}(X^{i,\mathrm{sig}}_{1:t_{\mathrm{P}}}, t) = \left\{\left(1, X^i_1\right), \ldots, \left(t_{\mathrm{P}}, X^i_{t_{\mathrm{P}}}\right), \left(t_{\mathrm{P}}+1, X^i_{t_{\mathrm{P}}}\right), \ldots, \left(t, X^i_{t_{\mathrm{P}}}\right)\right\}.$$

Given the dataset $\mathcal{D}^n_{\mathrm{surv}}$ and the landmark time t_{P}, the log-likelihood of the model can be defined as

$$\mathcal{L}(\theta \,|\, \mathcal{D}^n_{\mathrm{surv}}) = \sum_{i=1}^{n} \mathbb{1}_{T^i > t_{\mathrm{P}}}\Big(\Delta^i \log \lambda_\theta(T^i|X^i_{1:t_{\mathrm{P}}}, Z^i) - \int_1^{T^i} \lambda_\theta(u|X^i_{1:(u\wedge t_{\mathrm{P}})}, Z^i)du\Big).$$

Denoting by $\hat{\theta}_L$ a maximizer of this log-likelihood, we can form the predictions from the estimated survival function

$$S_{\hat{\theta}_L}(t_{\mathrm{P}} + \delta t|X^i_{1:t_{\mathrm{P}}}, Z^i) = \exp\Big(-\int_1^{t_{\mathrm{P}}+\delta t} \lambda_{\hat{\theta}_L}(u|X^i_{1:(u\wedge t_{\mathrm{P}})}, Z^i)du\Big).$$

RSF with landmark training. To extend the training of the RSF within the landmark setting, we follow a similar process to the RSF training outlined earlier, including the aggregation of longitudinal data. However, in this scenario, the longitudinal data is restricted to the period up to the landmark time t_{P}, meaning only the data $X^i_{1:t_{\mathrm{P}}}$ is considered for model training.

5.4 Performance evaluation

5.4.1 Dataset

Descriptive statistics. In this work, we select the historical order data of 1153 clients in the period of 2 years from 06-12-2021 to 12-11-2023. Any client that has not churned by 12-11-2023 is censored. In this dataset, 38.4% of the clients are censored. We extract from the historical order data 16 longitudinal features, which are computed on a one-week window and described below

- The dates on which the orders were placed
  - The total number of orders over the week (integer)
- The content of the orders
  - The number of items per order of the client, averaged over the week (integer)

  - The percentage margin over the week (float)
  - The refunds in euros following client complaints, summed over all orders of the week (float)
  - The weighted average price per kilo over the week (float)
  - The total sales from the client's orders over the week (float)
  - The average order value over the week (float)
  - The average discount percentage applied to all products bought over the week (float)
  - The number of distinct product high-level categories over the week (float)
  - The number of distinct product low-level categories over the week (float)
- The level of client satisfaction
  - The average client rating on the quality of products, overall orders of the week (float)
- The quality of the delivery
  - The percentage of products for which there was a complaint (float)
  - The number of missing items in the orders of the client, summed over all orders of the week (float)
  - The average client rating on the quality of delivery, and overall orders of the week (float)
  - The average client rating on the quality of client service, overall orders of the week (float)
  - The delivery delay indicator indicates whether there was a delivery delay at least once during the week (boolean integer)

Experiment setup Using the churn definition of four consecutive weeks without ordering, where a churned client who reorders is treated as a new client, we have a dataset of $n = 1823$ clients. After excluding features with more than 90 $\%$ missing data, $d = 14$ longitudinal features were selected. For weeks where clients placed no orders, all longitudinal measurements were filled with zero values for that week. Additionally, standardization was applied to the selected features before training. Figure 5.6 provides an example of two longitudinal features in both the calendar time and the client time, selected after preprocessing for the three clients A, B, and C.

The time point s_κ, which is used to split the preprocessed dataset into the training set and the testing set, is selected to be 22-05-2023. The training set then has a duration equal to one and a half years, starting from 06-12-2021, and includes a total of 1282 clients.

5.4.2 Evaluation metrics

In the real-time context, at a specified week, we would like to identify the top clients who have a higher risk of churn. We then evaluate the prediction performance of the model over two metrics which are the Time-Dependant Concordance Index and the Brier Score.

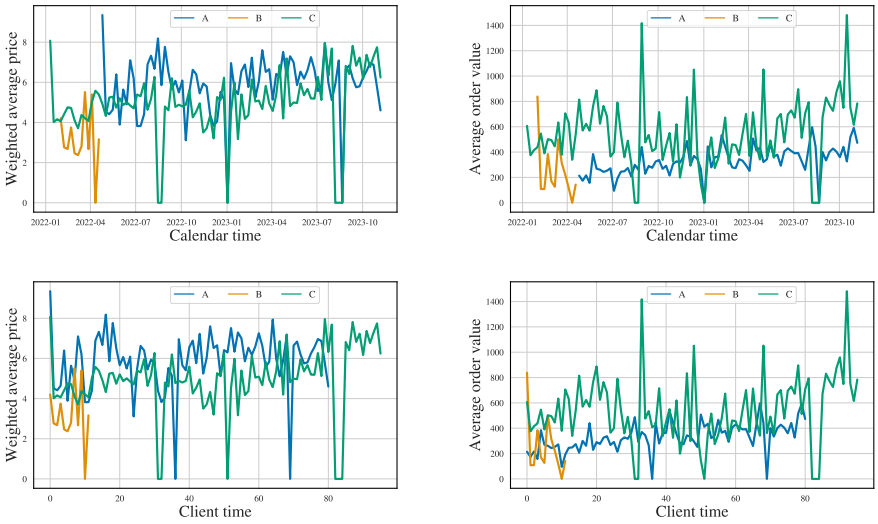

Figure 5.6 – Values of 2 different longitudinal features for 3 clients A, B, and C in both the calendar time (**top**) and the client time (**bottom**).

Time-Dependant Concordance Index. Following Lee et al. (2019), we measure the discriminative power of our models by using a time-dependent concordance index (or C-Index) $C(s_{\mathrm{P}}, \delta t)$ that captures our model's ability to correctly rank individuals on their predicted probability of survival. The concordance index $C(s_{\mathrm{P}}, \delta t)$ is then finally computed as

$$C(s_{\mathrm{P}},\, \delta_t) = \frac{\sum_{j=1}^{n}\sum_{i=1}^{n} \mathbb{1}_{\mathcal{R}^j(t_p^j,\delta_t)>\mathcal{R}^i(t_{\mathrm{p}}^i,\delta_t)} \mathbb{1}_{T^i>T^j,\, T^j\in[t_p^j,\, t_p^j+\delta t],\Delta^j=1}}{\sum_{j=1}^{n}\sum_{i=1}^{n} \mathbb{1}_{T^i>T^j,\, T^j\in[t_p^j,\, t_p^j+\delta t],\Delta^j=1}}.$$

This metric captures the capacity of the model to discriminate between client j and another client i through the probability of survival.

Brier Score. While the C-Index is a ranking-based measure, the Brier Score measures the accuracy in predictions by comparing the estimated survival function and the survival indicator function (Lee et al., 2019; Kvamme et al., 2019; Kvamme and Borgan, 2023). Formally, we define the Brier Score $\mathrm{BS}(s_{\mathrm{P}}, \delta t)$ as

$$\mathrm{BS}(s_{\mathrm{P}},\, \delta_t) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{T^i\in[t_{\mathrm{p}}^i,\, t_{\mathrm{p}}^i+\delta t],\Delta^i=1} r_\theta^i(t,\delta t)^2 + \mathbb{1}_{T^i>t_{\mathrm{p}}^i+\delta t}(1-r_\theta^i(t,\delta t))^2.$$

Contrarily to the C-index, the Brier Score is a measure of calibration of the predictions: it measures the distance between the estimated survival function and the indicator function of survival on the interval $[t_{\mathrm{P}}^i, t_{\mathrm{P}}^i + \delta t]$.

5.4.3 Results

In the following, we evaluate the prediction performance of the binary, temporal, and landmarking approaches using C-Index and Brier Score in different comparisons. First, we compare the results of the models within the binary approach. Next, we evaluate the performance of the models in the temporal approach and compare their performances with the best model from the binary approach. Finally, we examine the performance of the models in the landmark setting. In all the figures below, results are presented in terms of C-Index (*higher* is better) on the left and Brier Score (*lower* is better) on the right, with the boxplots for each model reflecting their prediction performance across multiple prediction weeks.

General performance of the binary approach. The experimental results in Figure 5.7 demonstrate the importance of using the signature transform to improve model performance in the binary approach, especially with the random forest classifier. Among the three feature engineering techniques evaluated - such as the combination of the last value function and the loyalty level, the combination of the last value function, the loyalty level and the lagged value function, or the signature transform - the application of the signature transform delivered the significant results both in C-Index and Brier Score metrics. This good performance highlights the ability of the signature transform to effectively capture the underlying dynamics of the data by summarising complex temporal dependencies that other feature engineering methods fail to achieve. However, the performance of signature-based logistic regression is not as strong, likely due to overfitting, even with careful regularisation. This suggests that logistic regression may lack the complexity required to handle the high-dimensional features generated by signature transformation. To mitigate this problem, possible solutions include applying dimensionality reduction, feature selection, or using a more complex model such as random forest or neural networks.

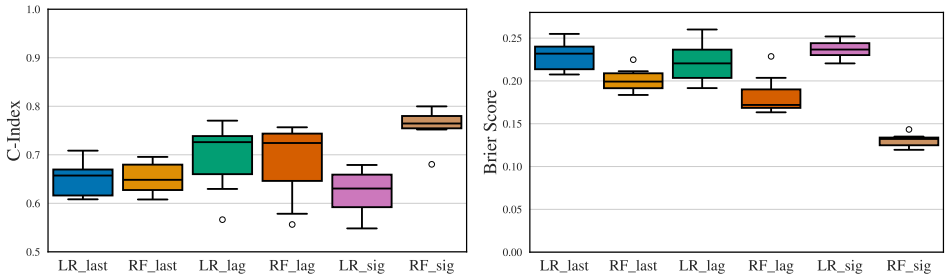

Figure 5.7 – Performance of the binary approach using two classifiers: logistic regression (**LR**) and random forest (**RF**), along with three sets of feature engineering techniques. The suffix **last** denotes the combination that uses the last value function and the level of loyalty, **lag** indicates the combination that uses the last value function, the level of loyalty, and lagged value function, and **sig** refers to the technique that employs signature transformation.

General performance of temporal approach. The experimental results in Figure 5.8 prove the effectiveness of our proposed CoxSig framework, which delivers strong perfor-

mance across both C-Index and Brier Score metrics, outperforming other methods, including the signature-based random forest—the best model from the binary approach. The signature transform continues to demonstrate its robust ability to capture the complexities of longitudinal data, significantly enhancing model performance not only within the Cox-based framework but also with the RSF. Although DDH performs well in terms of the Brier Score, its relatively weaker C-Index results indicate that it may not be the optimal choice for this churn prediction task that requires robust discrimination capabilities. On the other hand, our other proposed method, FLASH, does not perform as well in this churn prediction task, despite its ability to provide valuable insights into the importance of features affecting churn risk.

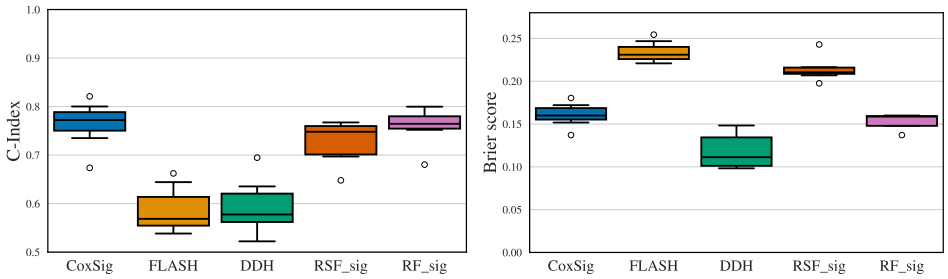

Figure 5.8 – Performance of the temporal approach using four frameworks: CoxSig, FLASH, DDH and RSF, in comparison with the signature-based random forest—the best model of the binary approach.

Model performance in the landmark setting. The experimental results in Figure 5.9 show the performance improvement achieved by training with the landmark setting, particularly in terms of Brier Score for both frameworks evaluated - CoxSig and RSF. While the non-landmark approach utilizes a full training set of 1282 clients, the landmark model operates with a significantly smaller median of 171 clients. RSF, in particular, demonstrates an improvement in both Brier Score and C-Index under the landmark setting. However, the CoxSig framework, while benefiting from a better Brier Score, shows a slight decrease in accuracy when evaluated by the C-Index metric. These results suggest that while the landmark setting generally enhances model performance, its impact on different metrics may vary depending on the framework used.

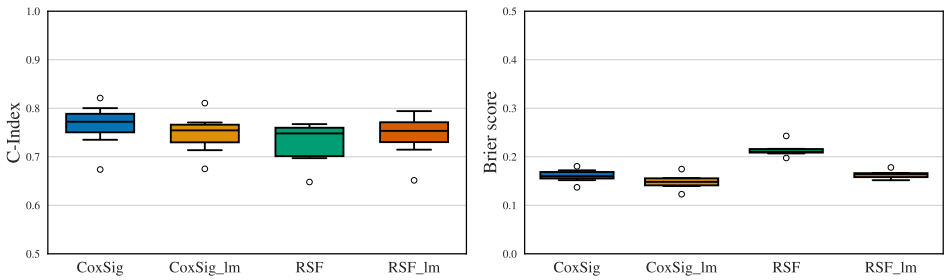

Figure 5.9 – The performance comparison between models trained with the landmark setting (with the suffix **lm**) and the standard setting, using two frameworks: CoxSig and RSF.

Model performance between new and old clients. The experimental results in Figure 5.10 highlight the variation in prediction performance between two distinct groups: new clients, whose first order is placed after the split date, and old clients, whose initial order occurs before the split date but continues to order afterward. This comparison provides valuable insights into how the model performs across different client groups based on their historical order data. Notably, the model demonstrates superior performance with old clients, as their extensive historical longitudinal data provides a stronger foundation for accurate predictions. This observation suggests that regularly retraining the model to incorporate more recent data is essential for maintaining predictive accuracy across all client segments.

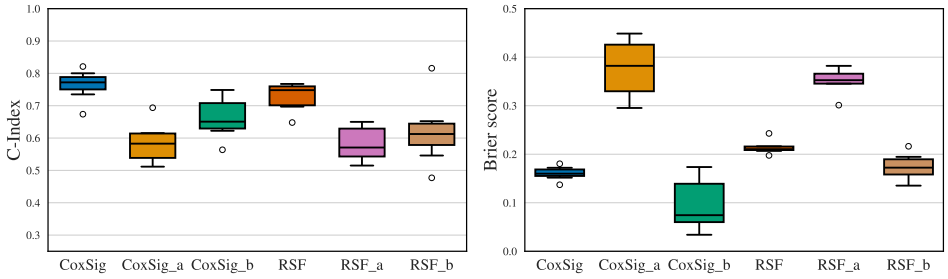

Figure 5.10 – The prediction performance comparison between new clients whose first date of order is after the split date (with the suffix **a**) and old clients whose first date of order is after the split date (with the suffix **b**), using two frameworks: CoxSig and RSF.

5.5 Conclusion

We developed and analyzed churn prediction models using three different approaches: binary classification, survival analysis, and an extension of the survival framework with landmarking. Overall, the CoxSig model demonstrated strong performance. Future research will focus on enhancing the interpretability of these models. Improving transparency around how specific longitudinal markers influence churn risk would offer the marketing team more actionable insights. This could involve creating new visualization tools to quantify the impact of individual features, making the model not only more accurate but also more user-friendly and practical for decision-making.

Conclusion

In this thesis, we have introduced and developed innovative frameworks for the growing field of survival analysis with longitudinal data, focusing on enhancing predictive performance and addressing high-dimensional challenges.

Chapter 3 presented a novel joint model that incorporates association features derived from longitudinal data directly into the survival model, without relying on assumptions in the longitudinal model. This approach has shown superior predictive performance and computational efficiency in comparison to existing state-of-the-art joint models, as evidenced by real-time numerical experiments conducted on both simulated and real datasets.

Chapter 4 extended this work by exploring the learning of individual-specific intensities of counting processes, driven by both static variables and longitudinal data. This framework represents the intensity function as the solution of a controlled differential equation, utilizing either neural or signature-based estimators. Through extensive experimental studies, the framework demonstrated its effectiveness and robustness in a variety of survival datasets.

In Chapter 5, we addressed the churn prediction problem at Califrais by applying three distinct methodologies: binary classification, survival analysis, and landmarking. We provided a comprehensive comparison of these approaches, highlighting their relative strengths and weaknesses in the real-time churn prediction context. The analysis demonstrated that the survival-based approaches, particularly the CoxSig model, offered superior performance in identifying both the likelihood and timing of churn, providing a more meaningful understanding of customer behavior.

Overall, this research contributes to the field by providing advanced tools for survival analysis and establishes a foundation for further developments aimed at fully integrating longitudinal data into survival models in various applied domains.

Future work offers several promising directions for extending and enhancing the contributions of this thesis.

First, the FLASH framework could be further developed by extending the implementation to accommodate more diverse types of longitudinal data. In particular, generalizing the Expectation-Maximisation algorithm to handle count or binary longitudinal features would make the model more applicable to a wider range of real-world contexts. In addition, the current assumption of fixed latent class membership could be relaxed to allow for dynamic class changes over time. This could be achieved by incorporating a Markov structure, al-

lowing the model to capture time-varying risk profiles, which would be particularly useful in contexts where individual risk levels fluctuate.

Second, improving the predictive ability of models such as NCDE or CoxSig remains an important focus for future work. By incorporating these models into a joint modeling framework, it would be possible to improve their predictive accuracy through the simultaneous modeling of both longitudinal and time-to-event data.

Third, extending the NCDE or CoxSig frameworks to applications involving counting processes could offer significant benefits across various sectors. For instance, predicting when clients will place their next order or when a patient is likely to be rehospitalized could significantly improve decision-making in sectors like retail and healthcare.

Finally, while the CoxSig model offers powerful predictive capabilities, it could benefit from efforts to improve its interpretability. Increasing the transparency of how specific longitudinal markers contribute to risk prediction would provide users - particularly those in fields such as healthcare and business - with more actionable insights. This could involve developing new visualization tools to quantify the impact of individual features, making the model not only more accurate but also more user-friendly and practical for decision-making.

Appendix A

Supplementary material of Chapter 3

Contents

To help the reader, Table A.1 provides a list of notations used in the paper.

Table A.1 – Summary of notation

Notation	Definition
X_i	Time-independent feature
Y_i	Longitudinal markers
T_i	Survival time
Δ_i	Censoring indication
g_i	Latent class membership variable
b_i	Random effects of the longitudinal markers
n	Number of training samples
p	Number of time-independent features
L	Number of longitudinal markers
K	Number of latent classes
D	Variance-covariance matrix of the b_i
U_i	Fixed-effect design matrix
V_i	Random-effect design matrix
ψ_i	Association features
ξ	Time-independent parameters
β	Fixed-effect parameter
γ	Joint association parameter
λ_0	Baseline hazard function
I_m	Identity matrix of $\mathbb{R}^{m\times m}$
$\mathbf{1}_m$	Vector of $\mathbb{R}^m$ having all coordinates equal to one
$\mathbf{0}_m$	Vector of $\mathbb{R}^m$ having all coordinates equal to zero
$\|\cdot\|_q$	The usual ℓ_q-quasi norm, $q > 0$

A.1 Details on the extended EM Algorithm

We detail in this section our learning methodology. First, recall that the penalized negative log-likelihood is defined by

$$\mathcal{L}_n^{\text{pen}}(\theta) = \mathcal{L}_n(\theta) + \sum_{k=1}^{K} \zeta_{1,k}\Omega_1(\xi_k) + \sum_{k=1}^{K} \zeta_{2,k}\Omega_2(\gamma_k), \tag{A.1}$$

where

$$\Omega_1(\xi_k) = (1-\eta)\left\|\xi_k\right\|_1 + \frac{\eta}{2}\left\|\xi_k\right\|_2^2 \quad \text{and} \quad \Omega_2(\gamma_k) = (1-\tilde{\eta})\left\|\gamma_k\right\|_1 + \tilde{\eta}\sum_{\ell=1}^{L}\left\|\gamma_k^\ell\right\|_2,$$

where the parameters $(\eta, \tilde{\eta}) \in [0,1]^2$ are fixed (depending on the expected level of sparsity), $\gamma_k^\ell = (\gamma_{k,1}^\ell, \ldots, \gamma_{k,M}^\ell)^\top \in \mathbb{R}^M$ is the subset of γ_k corresponding to the longitudinal marker ℓ, $\|\cdot\|_1$ (resp. $\|\cdot\|_2$) denotes the usual ℓ_1 (resp. ℓ_2) norm. In all our experiments, we take $\eta = 0.1$ and $\tilde{\eta} = 0.9$.

The goal is to minimize this objective function by the EM algorithm. This is done in two steps: compute the expectation of the negative complete log-likelihood with respect to the unobserved quantities (the random effects b_i and the latent classes g_i), then minimize the obtained quantity with respect to all parameters of the model in θ. For simplicity, we won't compute all terms of the expectation in the E-step but only the quantities used in the M-step. Moreover, we perform minimization with respect to θ in several steps, minimizing with respect to the block of parameters separately to obtain tractable updates.

A.1.1 E-step

Recall that, under our assumptions, the negative complete log-likelihood writes

$$\mathcal{L}_n^{\text{comp}}(\theta) = -n^{-1}\sum_{i=1}^{n}\Big(\log f_\theta(b_i) + \sum_{k=1}^{K}\mathbb{1}_{\{g_i=k\}}\big(\log \mathbb{P}_\theta(g_i=k) + \log f_\theta(Y_i \,|\, b_i, g_i=k) + \log f_\theta(T_i,\Delta_i \,|\, Y_i, g_i=k)\big)\Big).$$

Let us introduce a few matrix notations. Concatenating all longitudinal markers and all observation times, the mean of the vector $Y_i \,|\, b_i, g_i = k$ (defined in (3.3) in the main paper) can be rewritten $M_{ik} = U_i\beta_k + V_i b_i$, where we introduce the design matrices

$$U_i = \begin{bmatrix} U_i^1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & U_i^L \end{bmatrix} \in \mathbb{R}^{n_i \times q} \qquad \text{and} \qquad V_i = \begin{bmatrix} V_i^1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & V_i^L \end{bmatrix} \in \mathbb{R}^{n_i \times r}$$

and for all $\ell \in \{1,\dots,L\}$, one writes

$$U_i^\ell = \begin{bmatrix} u^\ell(t_{i1}^\ell)^\top \\ \vdots \\ u^\ell(t_{in_i^\ell}^\ell)^\top \end{bmatrix} \in \mathbb{R}^{n_i^\ell \times q_\ell} \qquad \text{and} \qquad V_i^\ell = \begin{bmatrix} v^\ell(t_{i1}^\ell)^\top \\ \vdots \\ v^\ell(t_{in_i^\ell}^\ell)^\top \end{bmatrix} \in \mathbb{R}^{n_i^\ell \times r_\ell}.$$

Under all assumptions of Section 3.2 in the main paper, we can then write explicitly the different terms. The random effects simply follow a Gaussian distribution, which yields

$$\log f_\theta(b_i) = -\frac{1}{2}(r\log 2\pi + \log\det(D) + b_i^\top D^{-1} b_i).$$

Then, the conditional density of the longitudinal features (in the Gaussian case) writes

$$\log f_\theta(Y_i \,|\, b_i, g_i=k) = -\frac{1}{2}\big(n_i\log 2\pi + \log\det(\Sigma_i) + (Y_i - U_i\beta_k - V_i b_i)^\top \Sigma_i^{-1}(Y_i - U_i\beta_k - V_i b_i)\big).$$

Finally, the survival terms in the complete likelihood write

$$\log f_\theta(T_i, \Delta_i \,|\, Y_i, g_i = k) = \Delta_i\big(\log \lambda_0(T_i) + \psi_i(T_i)^\top \gamma_k\big) - \sum_{j=1}^{J} \lambda_0(\tau_j) \exp\big(\psi_i(\tau_j)^\top \gamma_k\big) \mathbb{1}_{\{\tau_j \le T_i\}}.$$

We can therefore decompose the expected negative complete log-likelihood as

$$\begin{aligned}
\mathcal{Q}_n(\theta, \theta^{(w)}) &= \mathbb{E}_{\theta^{(w)}}[\mathcal{L}_n^{\text{comp}}(\theta) \,|\, \mathcal{D}_n] \\
&= -n^{-1} \sum_{i=1}^{n} \Big(A_i^1(D) + \sum_{k=1}^{K} \mathbb{P}_{\theta^{(w)}}(g_i = k | \mathcal{D}_n)\big(A_{i,k}^2(\xi) + A_i^3(\beta_k, \Sigma_i) + A_i^4(\gamma_k, \lambda_0)\big)\Big) \\
&\quad + \text{constants},
\end{aligned}$$

where, for any $b \in \mathbb{R}^r$, $D \in \mathbb{R}^{r \times r}$, $\beta \in \mathbb{R}^q$, $\Sigma \in \mathbb{R}^{n_i}$, we define

$$\begin{aligned}
A_i^1(D) &= \frac{1}{2}\mathbb{E}_{\theta^{(w)}}\big[\log\det(D) + b_i^\top D^{-1} b_i \,|\, \mathcal{D}_n\big], \\
A_{i,k}^2(\xi) &= \log\Big(\frac{e^{X_i^\top \xi_k}}{\sum_{j=1}^K e^{X_i^\top \xi_j}}\Big), \\
A_i^3(\beta, \Sigma) &= \frac{1}{2}\mathbb{E}_{\theta^{(w)}}\big[\log\det(\Sigma) + (Y_i - U_i\beta - V_i b_i)^\top \Sigma^{-1}(Y_i - U_i\beta - V_i b_i) \,|\, \mathcal{D}_n\big], \\
A_i^4(\gamma, \lambda_0) &= \Delta_i\big(\log \lambda_0(T_i) + \psi_i(T_i)^\top \gamma\big) - \sum_{j=1}^{J} \lambda_0(\tau_j) \exp\big(\psi_i(\tau_j)^\top \gamma\big) \mathbb{1}_{\{\tau_j \le T_i\}}.
\end{aligned}$$

As explained before, we do not compute the expectation of the negative complete likelihood but only of the quantities needed in the M-step. First, the following technical lemmas will prove useful in the E-step and makes use of conjugate properties of Gaussian distributions.

Lemma 2. For any $i \in \{1, \dots, n\}$, $k \in \{1, \dots K\}$,

$$Y_i \,|\, g_i = k \sim \mathcal{N}\big(U_i\beta_k, V_i D V_i^\top + \Sigma_i\big) \quad \text{and} \quad b_i \,|\, Y_i, g_i = k \sim \mathcal{N}\big(O_{i,k}, W_i\big),$$

where

$$O_{i,k} = W_i V_i^T {\Sigma_i}^{-1}(Y_i - U_i\beta_k) \quad \text{and} \quad W_i = \big(V_i^\top {\Sigma_i}^{-1} V_i + D^{-1}\big)^{-1}.$$

Proof. From (3.3) in the main paper we know that $Y_i \,|\, b_i, g_i = k \sim \mathcal{N}\big(M_{ik}, \Sigma_i\big)$ and $b_i \sim \mathcal{N}\big(0, D\big)$, which gives

$$Y_i | g_i = k \sim \mathcal{N}\big(U_i\beta_k, V_i D V_i^\top + \Sigma_i\big).$$

Moreover, by Bayes's rule, the distribution of $b_i | Y_i, g_i = k$ can be written

$$\begin{aligned}
f_\theta(b_i | Y_i, g_i = k) &\propto f_\theta(Y_i | b_i, g_i = k) f_\theta(b_i | g_i = k) \\
&\propto \exp\big((Y_i - U_i\beta_k - V_i b_i)^\top \Sigma_i^{-1}(Y_i - U_i\beta_k - V_i b_i) + b_i^\top D^{-1} b_i\big) \\
&\propto \exp\big((b_i - O_{i,k})^\top W_i^{-1}(b_i - O_{i,k})\big),
\end{aligned}$$

where $O_{i,k} = W_i {V_i^T \Sigma_i}^{-1}(Y_i - U_i\beta_k)$ and $W_i = \left(V_i^\top \Sigma_i^{-1} V_i + D^{-1}\right)^{-1}$. We then have

$$b_i | Y_i, g_i = k \sim \mathcal{N}\big(O_{i,k}, W_i\big).$$

The following lemma gives the three expectations that appear in this E-step.

Lemma 3. For any $i \in \{1, \dots, n\}$, $k \in \{1, \dots, K\}$, $\theta \in \mathbb{R}^P$, the three following integrals are closed-form and write

$$\mathbb{E}_\theta[b_i | \mathcal{D}_n] = \frac{\sum_{j=1}^K \mathbb{P}_\theta(g_i = j) f_\theta(T_i, \Delta_i | Y_i, g_i = j) f_\theta(Y_i | g_i = j) O_{i,j}}{\sum_{j=1}^K \mathbb{P}_\theta(g_i = j) f_\theta(T_i, \Delta_i | Y_i, g_i = j) f_\theta(Y_i | g_i = j)}, \tag{A.2}$$

$$\mathbb{E}_\theta[b_i b_i^\top | \mathcal{D}_n] = \frac{\sum_{j=1}^K \mathbb{P}_\theta(g_i = j) f_\theta(T_i, \Delta_i | Y_i, g_i = j) f_\theta(Y_i | g_i = j) (W_i + O_{i,j} {O_{i,j}}^\top)}{\sum_{j=1}^K \mathbb{P}_\theta(g_i = j) f_\theta(T_i, \Delta_i | Y_i, g_i = j) f_\theta(Y_i | g_i = j)}, \tag{A.3}$$

and

$$\tilde{\pi}_{ik}^\theta = \mathbb{P}_\theta(g_i = k | \mathcal{D}_n) = \frac{\mathbb{P}_\theta(g_i = k) f_\theta(T_i, \Delta_i | Y_i, g_i = k) f_\theta(Y_i | g_i = k)}{\sum_{j=1}^K \mathbb{P}_\theta(g_i = j) f_\theta(T_i, \Delta_i | Y_i, g_i = j) f_\theta(Y_i | g_i = j)}, \tag{A.4}$$

where $f_\theta(Y_i | g_i = j)$ is the density of the multivariate Gaussian distribution of Lemma 2.

Proof. By Assumption 2 in the main paper, the distribution of b_i given the observed data $\mathcal{D}_n$, for any $\theta \in \mathbb{R}^P$, writes

$$\begin{aligned} f_\theta(b_i | \mathcal{D}_n) &= \frac{f_\theta(b_i, T_i, \Delta_i, Y_i)}{f_\theta(T_i, \Delta_i, Y_i)} \\ &= \frac{1}{f_\theta(T_i, \Delta_i, Y_i)} \sum_{j=1}^K \mathbb{P}_\theta(g_i = j) f_\theta(b_i, T_i, \Delta_i, Y_i | g_i = j) \\ &= \frac{1}{f_\theta(T_i, \Delta_i, Y_i)} \sum_{j=1}^K \mathbb{P}_\theta(g_i = j) f_\theta(Y_i | g_i = j) f_\theta(T_i, \Delta_i | Y_i, g_i = j) f_\theta(b_i | Y_i, g_i = j). \end{aligned}$$

Similarly, we have

$$f_\theta(T_i, \Delta_i, Y_i) = \sum_{j=1}^K \mathbb{P}_\theta(g_i = j) f_\theta(Y_i | g_i = j) f_\theta(T_i, \Delta_i | Y_i, g_i = j).$$

This gives that, for any function μ,

$$\begin{aligned}\mathbb{E}_\theta[\mu(b_i)|\mathcal{D}_n] &= \int_{\mathbb{R}^r} \mu(b_i) f_\theta(b_i|\mathcal{D}_n)\mathrm{d}b_i \\ &= \frac{1}{f_\theta(T_i,\Delta_i,Y_i)} \sum_{j=1}^{K} \mathbb{P}_\theta(g_i=j) f_\theta(Y_i|g_i=j) f_\theta(T_i,\Delta_i|Y_i,g_i=j)\mathbb{E}_\theta[\mu(b_i)|Y_i,g_i=j].\end{aligned}$$

Hence, for the functions $\mu(b_i) = b_i$ and $\mu(b_i) = b_i b_i^\top$, we obtain the result by applying Lemma 2, which gives

$$\mathbb{E}_\theta[b_i|Y_i,g_i=j] = O_{i,j}, \quad \text{and} \quad \mathbb{E}_\theta[b_i b_i^\top|Y_i,g_i=j] = W_i + O_{i,j}{O_{i,j}}^\top.$$

In the same manner, we can see that for any $k \in \{1,\dots,K\}$,

$$\begin{aligned}\mathbb{E}_\theta[\mathbb{1}_{\{g_i=k\}}|T_i,\Delta_i,Y_i] &= \mathbb{P}_\theta(g_i=k|T_i,\Delta_i,Y_i) \\ &= \frac{1}{f_\theta(T_i,\Delta_i,Y_i)}\mathbb{P}_\theta(g_i=k) f_\theta(T_i,\Delta_i|Y_i,g_i=k) f_\theta(Y_i|g_i=k),\end{aligned}$$

which concludes the proof.

Extensions of the longitudinal model To support other types of longitudinal features (e.g., count, binary, etc) where the conditional longitudinal features $Y_i \,|\, b_i, g_i = k$ are not Gaussian, the simple solution is to choose distributions such that the random effect b_i and $Y_i \,|\, b_i, g_i = k$ have conjugate distributions. There are several options, for example using the beta-binomial distribution for binary features or negative-binomial for count features. In this case, the expectation $\mathbb{E}_\theta[\mu(b_i)|Y_i,g_i=j]$ and the density $f_\theta(b_i|Y_i,g_i=j)$ can be computed in closed forms (Molenberghs et al., 2010) and the extension is trivial. If the distributions are not conjugate, some numerical integration methods can be used to compute $\mathbb{E}_\theta[\mu(b_i)|Y_i,g_i=j]$ and $f_\theta(b_i|Y_i,g_i=j)$ (see, e.g., Fabio et al., 2012).

A.1.2 M-step: closed-form updates

Now, we assume that we are at step $w+1$ of the algorithm, meaning that we have a current value $\theta^{(w)}$ of the parameters and we update it to get the new parameters $\theta^{(w+1)}$ by solving

$$\theta^{(w+1)} \in \underset{\theta\in\mathbb{R}^P}{\arg\min}\, \mathcal{Q}_n^{\text{pen}}(\theta,\theta^{(w)}). \tag{A.5}$$

Note that we update for the coordinates of $\theta^{(w)}$ in order, which is $(D^{(w+1)}, (\xi_k^{(w+1)})_{k\in\{1,\dots,K\}}, (\beta_k^{(w+1)})_{k\in\{1,\dots,K\}}, (\gamma_k^{(w+1)})_{k\in\{1,\dots,K\}}, \lambda_0^{(w+1)}, \phi^{(w+1)})$. Then, following this order, the update of later coordinates uses the latest update of the previous ones. The update of several coordinates of $\theta^{(w+1)}$ can be obtained in closed form.

Lemma 4. At step $w+1$ of the EM algorithm, the update of D is

$$D^{(w+1)} = n^{-1} \sum_{i=1}^{n} \mathbb{E}_{\theta^{(w)}}[b_i b_i^\top | T_i, \Delta_i, Y_i],$$

where $\mathbb{E}_{\theta^{(w)}}[b_i b_i^\top | T_i, \Delta_i, Y_i]$ is given by (A.3).

Proof. The update for $D^{(w)}$ requires to solve the following minimization problem

$$D^{(w+1)} \in \underset{D \in \mathbb{R}^{q \times q}}{\arg\min} \; - n^{-1} \sum_{i=1}^{n} A_i^1(D). \tag{A.6}$$

We have for any $i \in \{1, \dots, n\}$,

$$A_i^1(D) = \mathbb{E}_{\theta^{(w)}}\big[\log\det(D) + b_i^\top D^{-1} b_i \big] = \log\det(D) + \int_{\mathbb{R}^r} b_i^\top D^{-1} b_i f_{\theta^{(w)}}(b_i | T_i, \Delta_i, Y_i) \mathrm{d}b_i.$$

The gradient of A_i^1 is here given by

$$\begin{aligned} \frac{\partial A_i^1(D)}{\partial D} &= D^{-\top} - \int_{\mathbb{R}^r} D^{-\top} b_i b_i^\top D^{-\top} f_{\theta^{(w)}}(b_i | T_i, \Delta_i, Y_i) \mathrm{d}b_i \\ &= D^{-\top} - D^{-\top} \mathbb{E}_{\theta^{(w)}}[b_i b_i^\top | T_i, \Delta_i, Y_i] D^{-\top}, \end{aligned}$$

where $D^{-\top}$ is the transpose of matrix D^{-1}. The proof is completed by cancelling the gradient, that is

$$-n^{-1} \sum_{i=1}^{n} \frac{\partial A_i^1(D)}{\partial D} = 0 \quad \Leftrightarrow \quad -D^{-\top} + n^{-1} \sum_{i=1}^{n} D^{-\top} \mathbb{E}_{\theta^{(w)}}[b_i b_i^\top | T_i, \Delta_i, Y_i] D^{-\top} = 0.$$

Lemma 5. At step $w+1$ of the EM algorithm, the update of β_k is

$${\beta_k}^{(w+1)} = \Big(\sum_{i=1}^{n} \tilde{\pi}_{ik}^{\theta^{(w)}} U_i^\top U_i \Big)^{-1} \Big(\sum_{i=1}^{n} \tilde{\pi}_{ik}^{\theta^{(w)}} U_i^\top (Y_i - V_i \mathbb{E}_{\theta^{(w)}}[b_i | T_i, \Delta_i, Y_i]) \Big), \tag{A.7}$$

where $\mathbb{E}_{\theta^{(w)}}[b_i | T_i, \Delta_i, Y_i]$ and $\tilde{\pi}_{ik}^{\theta^{(w)}}$ are given respectively by (A.2) and (A.4) .

Proof. The update for $\beta_k^{(w)}$ requires to solve the following minimization problem

$$\beta_k^{(w+1)} \in \underset{\beta \in \mathbb{R}^q}{\arg\min} \; - n^{-1} \sum_{i=1}^{n} \tilde{\pi}_{ik}^{\theta^{(w)}} A_i^3\big(\beta, \Sigma_i^{(w)}\big). \tag{A.8}$$

We have

$$\begin{aligned}
A_i^3(\beta, \Sigma_i^{(w)}) &= \frac{1}{2}\mathbb{E}_{\theta^{(w)}}\big[\log\det(\Sigma_i^{(w)}) + (Y_i - U_i\beta - V_i b_i)^\top {\Sigma_i^{(w)}}^{-1}(Y_i - U_i\beta - V_i b_i) \,|\, \mathcal{D}_n\big] \\
&= \frac{1}{2}\log\det(\Sigma_i^{(w)}) + \frac{1}{2}\mathbb{E}_{\theta^{(w)}}\big[(Y_i - U_i\beta - V_i b_i)^\top {\Sigma_i^{(w)}}^{-1}(Y_i - U_i\beta - V_i b_i) \,|\, \mathcal{D}_n\big] \\
&= \mathbb{E}_{\theta^{(w)}}\big[-(Y_i - V_i b_i)^\top {\Sigma_i^{(w)}}^{-1} U_i\beta + \frac{1}{2}\beta^\top U_i^\top {\Sigma_i^{(w)}}^{-1} U_i\beta \,|\, \mathcal{D}_n\big] + \text{constants} \\
&= -(Y_i - V_i\mathbb{E}_{\theta^{(w)}}\big[b_i|\,\mathcal{D}_n\big])^\top {\Sigma_i^{(w)}}^{-1} U_i\beta + \frac{1}{2}\beta^\top U_i^\top {\Sigma_i^{(w)}}^{-1} U_i\beta \; + \text{constants}.
\end{aligned}$$

(where we treat as constants all quantities independent of β). Then, the gradient of A_i^3 writes

$$\frac{\partial A_i^3(\beta, \Sigma_i^{(w)})}{\partial\beta} = -(Y_i - V_i\mathbb{E}_{\theta^{(w)}}[b_i|\mathcal{D}_n])^\top {\Sigma_i^{(w)}}^{-1} U_i + \beta^\top U_i^\top {\Sigma_i^{(w)}}^{-1} U_i.$$

Given the form of U_i and V_i along with the fact that $\Sigma_i^{(w)}$ is a diagonal matrix, we can rewrite the gradient of A_i^3 as

$$\frac{\partial A_i^3(\beta, \Sigma_i^{(w)})}{\partial\beta} = -(Y_i - V_i\mathbb{E}_{\theta^{(w)}}[b_i|\mathcal{D}_n])^\top U_i\tilde{\Sigma}^{(w)^{-1}} + \beta^\top U_i^\top U_i\tilde{\Sigma}^{(w)^{-1}},$$

where $\tilde{\Sigma}^{(w)}$ is the diagonal matrix whose diagonal is $({\phi_1^{(w)}\mathbf{1}_{q_1}}^\top, \ldots, {\phi_L^{(w)}\mathbf{1}_{q_L}}^\top)^\top \in \mathbb{R}^q$. The closed-form update of $\beta_k^{(w)}$ is then obtained by canceling the gradient, that is

$$\begin{aligned}
& -n^{-1}\sum_{i=1}^n \tilde{\pi}_{ik}^{\theta^{(w)}} \frac{\partial A_i^3(\beta, \Sigma_i^{(w)})}{\partial\beta} = 0 \\
\Leftrightarrow \quad & \sum_{i=1}^n \tilde{\pi}_{ik}^{\theta^{(w)}} \Big((Y_i - V_i\mathbb{E}_{\theta^{(w)}}[b_i|\mathcal{D}_n])^\top U_i\tilde{\Sigma}^{(w)^{-1}} - \beta^\top U_i^\top U_i\tilde{\Sigma}^{(w)^{-1}}\Big) = 0 \\
\Leftrightarrow \quad & \sum_{i=1}^n \tilde{\pi}_{ik}^{\theta^{(w)}} \Big((Y_i - V_i\mathbb{E}_{\theta^{(w)}}[b_i|\mathcal{D}_n])^\top U_i - \beta^\top U_i^\top U_i\Big) = 0 \\
\Leftrightarrow \quad & \beta = \Big(\sum_{i=1}^n \tilde{\pi}_{ik}^{\theta^{(w)}} U_i^\top U_i\Big)^{-1}\Big(\sum_{i=1}^n \tilde{\pi}_{ik}^{\theta^{(w)}} U_i^\top (Y_i - V_i\mathbb{E}_{\theta^{(w)}}[b_i|\mathcal{D}_n])\Big).
\end{aligned}$$

Lemma 6. At step $w+1$ of the EM algorithm, for any $j \in \{1, \ldots, J\}$, given the the update of $\lambda_0(\tau_j)$ is

$$\lambda_0^{(w+1)}(\tau_j) = \frac{\sum_{i=1}^n \Delta_i \mathbb{1}_{\{T_i=\tau_j\}}}{\sum_{i=1}^n \sum_{k=1}^K \tilde{\pi}_{ik}^{\theta^{(w)}} \exp\big(\psi_i(\tau_j)^\top \gamma_k^{(w+1)}\big)\mathbb{1}_{\{T_i\geq\tau_j\}}}, \tag{A.9}$$

where $\tilde{\pi}_{ik}^{\theta^{(w)}}$ is given by (A.4).

Proof. The update for $\lambda_0^{(w)}(\tau_j)$ requires to solve the minimization problem

$$\lambda_0^{(w+1)}(\tau_j) \in \underset{\lambda_0 \in \mathbb{R}}{\arg\min} \; - n^{-1} \sum_{i=1}^{n} \sum_{k=1}^{K} \tilde{\pi}_{ik}^{\theta^{(w)}} A_i^4(\gamma_k^{(w+1)}, \lambda_0). \tag{A.10}$$

We have

$$\begin{aligned}
&- n^{-1} \sum_{i=1}^{n} \sum_{k=1}^{K} \tilde{\pi}_{ik}^{\theta^{(w)}} A_i^4(\gamma_k^{(w+1)}, \lambda_0) \\
&\quad = -n^{-1} \sum_{i=1}^{n} \sum_{k=1}^{K} \tilde{\pi}_{ik}^{\theta^{(w)}} \Big(\Delta_i \big(\log \lambda_0(T_i) + \psi_i(T_i)^\top \gamma_k^{(w+1)} \big) \\
&\qquad\qquad - \sum_{j=1}^{J} \lambda_0(\tau_j) \exp\left(\psi_i(\tau_j)^\top \gamma_k^{(w+1)} \right) \mathbb{1}_{\{\tau_j \leq T_i\}} \Big) \\
&\quad = -n^{-1} \sum_{i=1}^{n} \Delta_i \log \lambda_0(T_i) \\
&\qquad\qquad + n^{-1} \sum_{i=1}^{n} \sum_{k=1}^{K} \tilde{\pi}_{ik}^{\theta^{(w)}} \sum_{j=1}^{J} \lambda_0(\tau_j) \exp\left(\psi_i(\tau_j)^\top \gamma_k^{(w+1)} \right) \mathbb{1}_{\{\tau_j \leq T_i\}} + \text{constants}
\end{aligned}$$

(where we keep only the terms with τ_j for any $j \in \{1, \ldots, J\}$). By taking the gradient of the previous expression over $\lambda_0(\tau_j)$ and setting it to zero, that is

$$-\frac{n^{-1}}{\lambda_0} \sum_{i=1}^{n} \Delta_i \mathbb{1}_{\{T_i = \tau_j\}} + n^{-1} \sum_{i=1}^{n} \sum_{k=1}^{K} \tilde{\pi}_{ik}^{\theta^{(w)}} \exp\left(\psi_i(\tau_j)^\top \gamma_k^{(w+1)} \right) \mathbb{1}_{\{T_i \geq \tau_j\}} = 0. \tag{A.11}$$

We then obtain the update for $\lambda_0^{(w)}(\tau_j)$ as the desired result.

Note that the closed-form update of $\lambda_0^{(w)}(\tau_j)$ is a Breslow-like estimator (Breslow, 1972) adapted to our model. Finally, recall that Σ_i is a diagonal matrix with a diagonal of the form $(\phi_1 \mathbf{1}_{n_i^1}{}^\top \cdots \phi_L \mathbf{1}_{n_i^L}{}^\top)$. Estimating Σ_i thus amounts to estimating $\phi_1, \ldots, \phi_L$, whose updates are given in the following lemma.

Lemma 7. At step $w+1$ of the EM algorithm, the update of ϕ_ℓ is

$$\begin{aligned}
\phi_\ell^{(w+1)} = \frac{1}{\sum_{i=1}^{n} n_i^\ell} \sum_{i=1}^{n} \sum_{k=1}^{K} \tilde{\pi}_{ik}^{\theta^{(w)}} \Big(& \big(Y_i^\ell - U_i^\ell \beta_k^{\ell(w+1)} \big)^\top \big(Y_i^\ell - U_i^\ell \beta_k^{\ell(w+1)} - 2 V_i^\ell \mathbb{E}_{\theta^{(w)}}[b_i^\ell \mid \mathcal{D}_n] \big) \\
& + \operatorname{Tr}\left(V_i^{\ell\top} V_i^\ell \mathbb{E}_{\theta^{(w)}}[b_i^\ell b_i^{\ell\top} \mid \mathcal{D}_n] \right) \Big),
\end{aligned} \tag{A.12}$$

where $\mathbb{E}_{\theta^{(w)}}[b_i^\ell \mid \mathcal{D}_n]$ and $\mathbb{E}_{\theta^{(w)}}[b_i^\ell b_i^{\ell\top} \mid \mathcal{D}_n]$ are obtained from (A.2) and (A.3).

Proof. The update for $\phi^{(w)}$ requires to solve the following minimization problem

$$\phi^{(w+1)} \in \underset{\phi \in \mathbb{R}^L}{\arg\min} \; - n^{-1} \sum_{i=1}^{n} \sum_{k=1}^{K} \tilde{\pi}_{ik}^{\theta^{(w)}} A_i^3(\beta_k^{(w+1)}, \mathrm{Diag}_i(\phi)), \tag{A.13}$$

where we denote by $\mathrm{Diag}_i(\phi)$ the diagonal matrix Σ_i to make clear its dependence on ϕ. We let ${M_{ik}}^{(w+1)} = U_i {\beta_k}^{(w+1)} + V_i b_i$, $M_{ik}^{\ell(w+1)} = U_i^\ell \beta_k^{\ell(w+1)} + V_i^\ell b_i^\ell$. Then, taking advantage of the structure of $\mathrm{Diag}_i(\phi)$, we have

$$\begin{aligned} A_i^3(\beta_k^{(w+1)}, \mathrm{Diag}_i(\phi)) &= \frac{1}{2} \log\det(\mathrm{Diag}_i(\phi)) + \mathbb{E}_{\theta^{(w)}}\left[(Y_i - M_{ik}^{(w+1)})^\top \mathrm{Diag}_i(\phi)^{-1} (Y_i - M_{ik}^{(w+1)}) \,|\, \mathcal{D}_n\right] \\ &= \frac{1}{2} \sum_{\ell=1}^{L} n_i^\ell \log \phi_\ell + \sum_{\ell=1}^{L} \frac{1}{\phi_\ell} \mathbb{E}_{\theta^{(w)}}\left[(Y_i^\ell - M_{ik}^{\ell(w+1)})^\top (Y_i^\ell - M_{ik}^{\ell(w+1)}) \,|\, \mathcal{D}_n\right]. \end{aligned}$$

Then, the gradient of A_i^3 along ϕ_ℓ is simply

$$\frac{\partial A_i^3(\beta_k^{(w+1)}, \mathrm{Diag}_i(\phi))}{\partial \phi_\ell} = \frac{n_i^\ell}{\phi_\ell} - \frac{1}{\phi_\ell^2} \mathbb{E}_{\theta^{(w)}}\left[\left(Y_i^\ell - M_{ik}^{\ell(w+1)}\right)^\top \left(Y_i^\ell - M_{ik}^{\ell(w+1)}\right) \,|\, \mathcal{D}_n\right],$$

The closed-form update of $\phi_\ell^{(w)}$ is then obtained by setting

$$\begin{aligned} & - n^{-1} \sum_{i=1}^{n} \sum_{k=1}^{K} \tilde{\pi}_{ik}^{\theta^{(w)}} \frac{\partial A_i^3(\beta_k^{(w+1)}, \mathrm{Diag}_i(\phi))}{\partial \phi_\ell} = 0 \\ \Leftrightarrow \quad & - n^{-1} \sum_{i=1}^{n} \sum_{k=1}^{K} \tilde{\pi}_{ik}^{\theta^{(w)}} \left(\frac{n_i^\ell}{\phi_\ell} - \frac{1}{\phi_\ell^2} \mathbb{E}_{\theta^{(w)}}\left[\left(Y_i^\ell - M_{ik}^{\ell(w+1)}\right)^\top \left(Y_i^\ell - M_{ik}^{\ell(w+1)}\right) \,|\, \mathcal{D}_n\right] \right) = 0 \\ \Leftrightarrow \quad & \sum_{i=1}^{n} \sum_{k=1}^{K} \tilde{\pi}_{ik}^{\theta^{(w)}} \left(n_i^\ell \phi_\ell^{(w+1)} - \mathbb{E}_{\theta^{(w)}}\left[\left(Y_i^\ell - M_{ik}^{\ell(w+1)}\right)^\top \left(Y_i^\ell - M_{ik}^{\ell(w+1)}\right) \,|\, \mathcal{D}_n\right] \right) = 0. \end{aligned}$$

The result follows from the fact that

$$\begin{aligned} & \mathbb{E}_{\theta^{(w)}}\left[\left(Y_i^\ell - M_{ik}^{\ell(w+1)}\right)^\top \left(Y_i^\ell - M_{ik}^{\ell(w+1)}\right) \,|\, \mathcal{D}_n\right] \\ & \quad = \mathbb{E}_{\theta^{(w)}}\left[\left(Y_i^\ell - U_i^\ell \beta_k^{\ell(w+1)}\right)^\top \left(Y_i^\ell - U_i^\ell \beta_k^{\ell(w+1)} - 2V_i^\ell b_i^\ell\right) + b_i^\top V_i^\top V_i b_i \,|\, \mathcal{D}_n\right] \\ & \quad = \left(Y_i^\ell - U_i^\ell \beta_k^{\ell(w+1)}\right)^\top \left(Y_i^\ell - U_i^\ell \beta_k^{\ell(w+1)} - 2V_i^\ell \mathbb{E}_{\theta^{(w)}}[b_i^\ell \,|\, \mathcal{D}_n]\right) + \mathrm{Tr}\left({V_i^\ell}^\top V_i^\ell \mathbb{E}_{\theta^{(w)}}[b_i^\ell b_i^{\ell\top} \,|\, \mathcal{D}_n]\right). \end{aligned}$$

A.1.3 M-step: Update ξ

In $\mathcal{Q}_n(\theta, \theta^{(w)})$, the parameter ξ appears only in the term

$$
\begin{aligned}
&- n^{-1} \sum_{i=1}^{n} \sum_{k=1}^{K} \tilde{\pi}_{ik}^{\theta^{(w)}} A_{i,k}^2(\xi) \\
&= -n^{-1} \sum_{i=1}^{n} \sum_{k=1}^{K} \tilde{\pi}_{ik}^{\theta^{(w)}} \log\big(\frac{e^{X_i^\top \xi_k}}{\sum_{j=1}^{K} e^{X_i^\top \xi_j}}\big) \\
&= -n^{-1} \sum_{i=1}^{n} \Big(\tilde{\pi}_{ik}^{\theta^{(w)}} \log\big(1 + \sum_{\substack{j \neq k \\ j=1}}^{K} e^{X_i^\top (\xi_j - \xi_k)}\big) \\
&\qquad + \sum_{\substack{m \neq k \\ m=1}}^{K} \tilde{\pi}_{im}^{\theta^{(w)}} \log\big(1 + e^{X_i^\top (\xi_k - \xi_m)} + \sum_{\substack{j \neq k, j \neq m \\ j=1}}^{K} e^{X_i^\top (\xi_j - \xi_m)}\big)\Big).
\end{aligned}
$$

For $k \in \{1, \ldots, K\}$, the update for $\xi_k^{(w)}$ requires to solve the minimization problem

$$
\xi_k^{(w+1)} \in \underset{\xi \in \mathbb{R}^p}{\arg\min}\, \mathcal{F}_{1,k}(\xi) + \zeta_{1,k} \Omega_1(\xi), \tag{A.14}
$$

where $\mathcal{F}_{1,k}$ is defined by

$$
\begin{aligned}
\mathcal{F}_{1,k}(\xi) = n^{-1} \sum_{i=1}^{n} \Big(&\tilde{\pi}_{ik}^{\theta^{(w)}} \log\big(1 + \sum_{\substack{j \neq k \\ j=1}}^{K} e^{X_i^\top (\xi_j - \xi)}\big) \\
&+ \sum_{\substack{m \neq k \\ m=1}}^{K} \tilde{\pi}_{im}^{\theta^{(w)}} \log\big(1 + e^{X_i^\top (\xi - \xi_m)} + \sum_{\substack{j \neq k, j \neq m \\ j=1}}^{K} e^{X_i^\top (\xi_j - \xi_m)}\big)\Big)
\end{aligned}
$$

and Ω_1 is the elastic net regularization. We choose to solve (A.14) using the L-BFGS-B algorithm (Zhu et al., 1997) which belongs to the class of quasi-Newton optimization routines and solves the given minimization problem by computing approximations of the inverse Hessian matrix of the objective function. It can deal with differentiable convex objectives with box constraints.

In order to use it with l_1 part of the elastic net regularization, which is not differentiable, we use the trick borrowed from Andrew and Gao (2007): for $a \in \mathbb{R}$, write $|a| = a^+ + a^-$, where a^+ and a^- are respectively the positive and negative part of a, and add the constraints $a^+ \geq 0$ and $a^- \geq 0$. Namely, we rewrite the minimization problem (A.14) as the following

differentiable problem with box constraints

$$\begin{aligned} &\text{minimize} && \mathcal{F}_{1,k}(\xi^+ - \xi^-) + \zeta_{1,k}\big((1-\eta)\sum_{j=1}^{p}(\xi_j^+ + \xi_j^-) + \frac{\eta}{2}\left\|\xi^+ - \xi^-\right\|_2^2\big) \\ &\text{subject to} && \xi_j^+ \geq 0 \text{ and } \xi_j^- \geq 0 \text{ for } j \in \{1,\dots,p\} \end{aligned} \tag{A.15}$$

where $\xi^{\pm} = (\xi_1^{\pm}, \dots, \xi_p^{\pm})^\top$. The L-BFGS-B solver requires the exact value of the gradient, which is easily given by

$$\frac{\partial \mathcal{F}_{1,k}(\xi)}{\partial \xi} = -n^{-1}\sum_{i=1}^{n}\big(\tilde{\pi}_{ik}^{\theta^{(w)}} - \frac{e^{X_i^\top \xi}}{e^{X_i^\top \xi} + \sum_{\substack{j\neq k \\ j=1}}^{K} e^{X_i^\top \xi_j}}\big)X_i^\top. \tag{A.16}$$

In practice, we use the `Python` solver `fmin_l_bfgs_b` from `scipy.optimize` (Virtanen et al., 2020).

A.1.4 M-step: Update γ

In $\mathcal{Q}_n(\theta, \theta^{(w)})$, for $k \in \{1,\dots,K\}$, γ_k appears only in the term

$$\begin{aligned} &-n^{-1}\sum_{i=1}^{n}\tilde{\pi}_{ik}^{\theta^{(w)}} A_i^4(\gamma_k, \lambda_0) \\ &= -n^{-1}\sum_{i=1}^{n}\tilde{\pi}_{ik}^{\theta^{(w)}}\Big(\Delta_i\big(\log\lambda_0(T_i) + \psi_i(T_i)^\top\gamma_k\big) - \sum_{j=1}^{J}\lambda_0(\tau_j)\exp\big(\psi_i(\tau_j)^\top\gamma_k\big)\mathbb{1}_{\{\tau_j \leq T_i\}}\Big) \\ &= -n^{-1}\sum_{i=1}^{n}\tilde{\pi}_{ik}^{\theta^{(w)}}\Big(\Delta_i\psi_i(T_i)^\top\gamma_k - \sum_{j=1}^{J}\lambda_0(\tau_j)\exp\big(\psi_i(\tau_j)^\top\gamma_k\big)\mathbb{1}_{\{\tau_j \leq T_i\}}\Big) + \text{constants}. \end{aligned}$$

Then the update for $\gamma_k^{(w)}$ requires to solve the following minimization problem

$$\gamma_k^{(w+1)} \in \underset{\gamma\in\mathbb{R}^{LM}}{\arg\min}\, \mathcal{F}_{2,k}(\gamma) + \zeta_{2,k}\Omega_2(\gamma), \tag{A.17}$$

where $\mathcal{F}_{2,k}$ is defined by

$$\mathcal{F}_{2,k}(\gamma) = -n^{-1}\sum_{i=1}^{n}\tilde{\pi}_{ik}^{\theta^{(w)}}\Big(\Delta_i\psi_i(T_i)^\top\gamma - \sum_{j=1}^{J}\lambda_0^{(w)}(\tau_j)\exp\big(\psi_i(\tau_j)^\top\gamma\big)\mathbb{1}_{\{\tau_j \leq T_i\}}\Big)$$

and Ω_2 is the sparse group lasso regularization. We choose to solve problem (A.17) using the iterative soft-thresholding algorithm (ISTA), which is a proximal gradient descent algorithm (Beck and Teboulle, 2009). In our context, this method requires the gradient of $\mathcal{F}_{2,k}$ as well as the proximal operator (Moreau, 1962) of the sparse group lasso. We refer to the

proof of Yuan et al. (2011, Theorem 1) to show that the proximal operator of the sparse group lasso can be expressed as the composition of the group lasso and the lasso proximal operators, which are both well known analytically (Bach et al., 2012) and tractable. The gradient of $\mathcal{F}_{2,k}$ is here given by

$$\frac{\partial \mathcal{F}_{2,k}(\gamma)}{\partial \gamma} = -n^{-1} \sum_{i=1}^{n} \tilde{\pi}_{ik}^{\theta^{(w)}} \Big(\Delta_i - \sum_{j=1}^{J} \lambda_0^{(w)}(\tau_j) \exp\left(\psi_i(\tau_j)^\top \gamma\right) \mathbb{1}_{\{T_i \geq \tau_j\}} \Big) \psi_i(T_i)^\top. \quad \text{(A.18)}$$

We use the `Python` library `copt` for the implementation of proximal gradient descent (Fabian Pedregosa, 2020) and we propose in our `FLASH` package a first `Python` implementation for the proximal operator of the sparse group lasso.

A.1.5 Convex optimization problems with respect to ξ and γ

Lemma 8. The optimization problems defined in (A.14) and (A.17) are convex.

Proof. Given first order derivative of $\mathcal{F}_{1,k}(\xi)$ in (A.16), we show that the second order derivative of $\mathcal{F}_{1,k}(\xi)$ is positive definite

$$\frac{\partial^2 \mathcal{F}_{1,k}(\xi)}{\partial \xi \partial \xi^\top} = n^{-1} \sum_{i=1}^{n} \Big(\frac{e^{X_i^\top \xi}}{e^{X_i^\top \xi} + \sum\limits_{\substack{j \neq k \\ j=1}}^{K} e^{X_i^\top \xi_j}} \Big) \Big(1 - \frac{e^{X_i^\top \xi}}{e^{X_i^\top \xi} + \sum\limits_{\substack{j \neq k \\ j=1}}^{K} e^{X_i^\top \xi_j}} \Big) X_i X_i^\top \in S_{++}^p.$$

Given first order derivative of $\mathcal{F}_{2,k}(\gamma)$ in (A.18), we show that the second order derivative of $\mathcal{F}_{2,k}(\gamma)$ is positive definite

$$\frac{\partial^2 \mathcal{F}_{2,k}(\gamma)}{\partial \gamma \partial \gamma^\top} = n^{-1} \sum_{i=1}^{n} \Big(\tilde{\pi}_{ik}^{\theta^{(w)}} \sum_{j=1}^{J} \lambda_0^{(w)}(\tau_j) \exp\left(\psi_i(\tau_j)^\top \gamma\right) \mathbb{1}_{\{T_i \geq \tau_j\}} \Big) \psi_i(T_i) \psi_i(T_i)^\top \in S_{++}^{LM}.$$

As we already defined the elastic net and sparse group lasso regularization

$$\Omega_1(\xi) = (1-\eta) \left\|\xi\right\|_1 + \frac{\eta}{2} \left\|\xi\right\|_2^2 \quad \text{and} \quad \Omega_2(\gamma) = (1-\tilde{\eta}) \left\|\gamma\right\|_1 + \tilde{\eta} \sum_{\ell=1}^{L} \left\|\gamma^\ell\right\|_2,$$

with $(\eta, \tilde{\eta}) \in [0,1]^2$. Note that every norm is convex and a non-negative weighted sum of convex functions is convex (Boyd and Vandenberghe, 2004) then $\Omega_1(\xi)$ and $\Omega_2(\gamma)$ are convex functions. Therefore, $\mathcal{F}_{1,k}(\xi) + \Omega_1(\xi)$ and $\mathcal{F}_{2,k}(\gamma) + \Omega_2(\gamma)$ are strictly convex.

A.1.6 The extended EM algorithm

Algorithm 1 below describes the main steps of our proposed EM algorithm.

Algorithm 1 The extended EM algorithm for FLASH inference

Data: Training data $\mathcal{D}_n$; tuning hyper-parameters $(\zeta_{1,k}, \zeta_{2,k})_{k\in\{1,\dots,K\}}$
Input: maximum iteration W, tolerance ε
Output: Last parameters $\hat{\theta} \in \mathbb{R}^P$

Initialize parameters $\theta^{(0)} \in \mathbb{R}^P$
for $w = 1, \dots, W$ **do**
  E-step:
    Compute $(\mathbb{E}_{\theta^{(w)}}[b_i|T_i, \Delta_i, Y_i])_{i\in\{1,\dots,n\}}$
    Compute $(\mathbb{E}_{\theta^{(w)}}[b_i b_i^\top|T_i, \Delta_i, Y_i])_{i\in\{1,\dots,n\}}$
    Compute $(\widetilde{\pi}_{ik}^{\theta^{(w)}})_{\substack{i\in\{1,\dots,n\}\\ k\in\{1,\dots,K\}}}$
  M-step:
    Update $D^{(w+1)}$
    Update $(\xi_k^{(w+1)})_{k\in\{1,\dots,K\}}$ with L-BFGS-B
    Update $(\beta_k^{(w+1)})_{k\in\{1,\dots,K\}}$
    Update $(\gamma_k^{(w+1)})_{k\in\{1,\dots,K\}}$ with proximal gradient descent
    Update $\lambda_0^{(w+1)}$ and $\phi^{(w+1)}$
    if $\left(\mathcal{L}_n^{\text{pen}}(\theta^{(w+1)}) - \mathcal{L}_n^{\text{pen}}(\theta^{(w)})\right)/\mathcal{L}_n^{\text{pen}}(\theta^{(w)}) < \varepsilon$ **then**
      break
    end if
end for
Return $\hat{\theta} = \theta^{(w+1)}$

A.1.7 Monotone convergence

By denoting $\zeta_p^{(w)} = (\theta_1^{(w+1)}, \dots, \theta_p^{(w+1)}, \theta_{p+1}^{(w)}, \dots, \theta_P^{(w)})$ and from the definition of our proposed EM algorithm, at step $w+1$, we have

$$\mathcal{Q}_n^{\text{pen}}(\theta^{(w)}, \theta^{(w)}) \geq \mathcal{Q}_n^{\text{pen}}(\zeta_1^{(w)}, \theta^{(w)}) \geq \cdots \geq \mathcal{Q}_n^{\text{pen}}(\zeta_{P-1}^{(w)}, \theta^{(w)}) \geq \mathcal{Q}_n^{\text{pen}}(\theta^{(w+1)}, \theta^{(w)}). \quad \text{(A.19)}$$

Then, we are in a generalized EM (GEM) setting (Dempster et al., 1977), where

$$\mathcal{Q}_n^{\text{pen}}(\theta^{(w+1)}, \theta^{(w)}) \leq \mathcal{Q}_n^{\text{pen}}(\theta^{(w)}, \theta^{(w)}). \quad \text{(A.20)}$$

For such algorithms, we refer to monotonicity of the likelihood property from the book of McLachlan and Krishnan (2007, Section 3.3) to show that the objective function (A.1) decreases at each iteration, namely

$$\mathcal{L}_n^{\text{pen}}(\theta^{(w+1)}) \leq \mathcal{L}_n^{\text{pen}}(\theta^{(w)}).$$

A.2 Mathematical details of JLCMs and SREMs

Given our notation used in the main body of the paper, we define here the sub-models of the two main approaches of joint models: JLCMs and SREMs.

JLCMs It assumes that the population is heterogeneous and that there are homogeneous latent classes that share the same marker trajectories and the same prognosis. The latent class membership probability is assumed to take the form of multinomial logistic regression

$$\mathbb{P}[g_i = k \mid X_i] = \frac{e^{X_i^\top \xi_k}}{\sum_{j=1}^{K} e^{X_i^\top \xi_j}}.$$

The dependence between the time-to-event and the longitudinal marker is fully captured by a latent class structure. There are no shared associations between the longitudinal and survival models. Given the latent class membership, each submodel is assumed to be independent. If we choose Gaussian linear model for longitudinal markers and Cox relative risk model for the time-to-event, we have

$$y_i^\ell(t_{ij}^\ell) \mid b_i^\ell,\, g_i = k \sim \mathcal{N}(m_{ik}^\ell(t_{ij}^\ell), \phi_\ell) \quad \text{and} \quad \lambda(t \mid g_i = k) = \lambda_0(t) \exp\Big(X_i^\top \gamma_k\Big),$$

where γ_k is the p-vector of unknown parameters. We consider the implementation of JLCMs in R package `LCMM` (function `mpjlcmm`). In this context, the predictive marker for subject i at time s_i is

$$\widehat{\mathcal{R}}_{ik}(s_i) = \frac{\mathbb{P}_{\hat\theta}(g_i = k) f_{\hat\theta}(T_i = s_i, \Delta_i = 0, \mathcal{Y}_i(s_i^-) | g_i = k)}{\sum_{j=1}^{K} \mathbb{P}_{\hat\theta}(g_i = j) f_{\hat\theta}(T_i = s_i, \Delta_i = 0, \mathcal{Y}_i(s_i^-) | g_i = j)},$$

where the density $f_{\hat\theta}$ are the one corresponding to a JLCM model.

SREMs It assumes a homogeneous population of subjects and the dependency between the time-to-event and the longitudinal marker is influenced by some random effects learned in a linear mixed model. If we choose Gaussian linear model for longitudinal markers, we have

$$y_i^\ell(t_{ij}^\ell) \mid b_i^\ell \sim \mathcal{N}(m_i^\ell(t_{ij}^\ell), \phi_\ell),$$

where $m_i^\ell(t_{ij}^\ell) = u^\ell(t_{ij}^\ell)^\top \beta^\ell + v^\ell(t_{ij}^\ell)^\top b_i^\ell$ and β^ℓ is a r_ℓ-vector of unknown fixed effect parameters. The random effects are included as covariates in the survival model through the shared association functions ϕ. If we choose Cox relative risk model for the time-to-event, we have

$$\lambda(t \mid g_i = k) = \lambda_0(t) \exp\Big(X_i^\top \gamma_0 + \sum_{\ell=1}^{L} \phi(b_i^\ell, t_i)^\top \gamma^\ell\Big),$$

where γ_0 and γ^ℓ are unknown parameters.

We consider the implementation of SREMs in R package `JMbayes`. In this context, the predictive marker for subject i at time s_i is

$$\widehat{\mathcal{R}}_i(s_i) = \exp\Big(X_i^\top \gamma_0 + \sum_{\ell=1}^{L} \phi(b_i^\ell, s_i)^\top \gamma^\ell\Big),$$

where the shared associations takes the form $\phi(b_i^\ell, s_i) = u^\ell(s_i)^\top \beta^\ell + v^\ell(s_i)^\top b_i^\ell$.

A.3 Experimental details and additional experiments

A.3.1 Initialization

Initialization

In order to help convergence, $\theta^{(0)}$ should be well chosen. We then give some details about the starting point $\theta^{(0)}$ of this algorithm. For all $k = 1, \ldots, K$, we first choose $\xi_k^{(0)} = \mathbf{0}_d$ and ${\gamma_k}^{(0)} = 0.01 * \mathbf{1}_{LA}$. Then, we initialize $\lambda_0^{(0)}$ like if there are no latent classes ($\gamma_1^{(0)} = \cdots = \gamma_K^{(0)}$) with a standard Cox proportional hazards regression with time-independent features. Finally, the longitudinal submodel parameters $\beta_k^{(0)}$, $D^{(0)}$ and $\phi^{(0)}$ are initialized – again like if there are no latent classes ($\beta_1^{(0)} = \cdots = \beta_K^{(0)}$) – using a multivariate linear mixed model with an explicit EM algorithm, being itself initialized with univariate fits.

Multivariate linear mixed model

Let us derive here the explicit EM algorithm for the multivariate Gaussian linear mixed model used to initialize the longitudinal parameters $\beta_k^{(0)}$, $D^{(0)}$ and $\phi^{(0)}$ in the proposed EM algorithm in Section A.3.1, acting as if there are no latent classes ($\beta_1^{(0)} = \cdots = \beta_K^{(0)}$). For the sake of simplicity, let us denote here

$$\theta = (\beta^\top, D, \phi^\top)^\top \in \mathbb{R}^P$$

the parameter vector to infer. The conditional distribution of $Y_i|b_i$ then writes

$$f(Y_i|b_i;\theta) = -(2\pi)^{-\frac{n_i}{2}} \det(\Sigma_i)^{-\frac{1}{2}} \exp^{-\frac{1}{2}(Y_i - M_i)^\top \Sigma_i^{-1}(Y_i - M_i)},$$

where $M_i = U_i\beta + V_i b_i$ in this context. The negative complete log-likelihood then writes

$$\begin{aligned}\mathcal{L}_n^{\text{comp}}(\theta) &= \mathcal{L}_n^{\text{comp}}(\theta; \mathcal{D}_n, \boldsymbol{b}) \\ &= \sum_{i=1}^{n} -\frac{1}{2}\big(n_i \log 2\pi + \log\det(\Sigma_i) + (Y_i - M_i)^\top \Sigma_i^{-1}(Y_i - M_i)\big) \\ &\qquad - \frac{1}{2}\big(r\log 2\pi + \log\det(D) + b_i^\top D^{-1} b_i\big).\end{aligned}$$

E-step. Supposing that we are at step $w+1$ of the algorithm, with current iterate denoted $\theta^{(w)}$, we need to compute the expectation of the negative complete log-likelihood conditional on the observed data and the current estimate of the parameters, which is given by

$$\mathcal{Q}_n(\theta, \theta^{(w)}) = \mathbb{E}_{\theta^{(w)}}[\mathcal{L}_n^{\text{comp}}(\theta)|\mathcal{D}_n].$$

Here, the calculation of this quantity is reduced to the calculation of $\mathbb{E}_{\theta^{(w)}}[b_i|Y_i]$ and $\mathbb{E}_{\theta^{(w)}}[b_i b_i^T|Y_i]$ for $i = 1, \dots, n$. The marginal distributions of Y_i and b_i being both Gaussian, one has from Bayes Theorem

$$f(b_i|Y_i; \theta^{(w)}) \propto \exp\big(-\frac{1}{2}(b_i - \mu_i^{(w)})^\top {\Omega_i^{(w)}}^{-1}(b_i - \mu_i^{(w)})\big)$$

where

$$\Omega_i^{(w)} = ({V_i}^\top {\Sigma_i^{(w)}}^{-1} V_i + {D^{(w)}}^{-1})^{-1} \quad \text{and} \quad \mu_i^{(w)} = \Omega_i^{(w)} {V_i}^\top {\Sigma_i^{(w)}}^{-1}(Y_i - U_i\beta^{(w)}).$$

Then, one has

$$\begin{cases}\mathbb{E}_{\theta^{(w)}}[b_i|Y_i] = \mu_i^{(w)}, \\ \mathbb{E}_{\theta^{(w)}}[b_i b_i^T|Y_i] = \Omega_i^{(w)} + \mu_i^{(w)} {\mu_i^{(w)}}^\top.\end{cases}$$

M-step. Here, we need to compute

$$\theta^{(w+1)} \in \underset{\theta \in \mathbb{R}^P}{\arg\min}\, \mathcal{Q}_n(\theta, \theta^{(w)}).$$

The parameters updates are then naturally given in closed form by zeroing the gradient. One obtains

$$\beta^{(w+1)} = \Big(\sum_{i=1}^{n} U_i^\top U_i\Big)^{-1}\Big(\sum_{i=1}^{n} U_i^\top Y_i - U_i V_i \mathbb{E}_{\theta^{(w)}}[b_i|Y_i]\Big),$$

$$\begin{aligned}\phi_\ell^{(w+1)} = \Big(\sum_{i=1}^{n} n_i^\ell\Big)^{-1}\Big(\sum_{i=1}^{n} &\big(Y_i^\ell - U_i^\ell \beta_\ell^{(w+1)}\big)^\top \big(Y_i^\ell - U_i^\ell \beta_\ell^{(w+1)} - 2V_i^\ell \mathbb{E}_{\theta^{(w)}}[b_i^\ell|Y_i^\ell]\big) \\ &+ \operatorname{Tr}\big({V_i^\ell}^\top V_i^\ell \mathbb{E}_{\theta^{(w)}}[b_i^\ell {b_i^\ell}^\top|Y_i^\ell]\big)\Big)\end{aligned}$$

and

$$D^{(w+1)} = n^{-1} \sum_{i=1}^{n} \mathbb{E}_{\theta^{(w)}}[b_i {b_i}^\top | Y_i].$$

Implementation of multivariate linear mixed model

We implement an EM algorithm for fitting a multivariate linear mixed model used to initialize parameters of longitudinal submodel in Algorithm 1. Let us introduce the list $\Omega^{(w)} = [\Omega_1^{(w)}, \ldots, \Omega_n^{(w)}]$, the matrices $\mu = [\mu_1, \ldots, \mu_n] \in \mathbb{R}^{r \times n}$, $U^\ell = [{U_1^\ell}^\top \cdots {U_n^\ell}^\top]^\top \in \mathbb{R}^{n_\ell \times q_\ell}$, $U = [U_1^\top \cdots U_n^\top]^\top \in \mathbb{R}^{\mathcal{N} \times q}$,

$$V^\ell = \begin{bmatrix} V_1^\ell & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & V_n^\ell \end{bmatrix}, \quad V = \begin{bmatrix} V_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & V_n \end{bmatrix}, \quad \Omega^{\ell(w)} = \begin{bmatrix} \Omega_1^{\ell\,(w)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \Omega_n^{\ell\,(w)} \end{bmatrix}$$

that belong respectively in $\mathbb{R}^{n_\ell \times nr_\ell}$, $\mathbb{R}^{\mathcal{N} \times nr}$ and $\mathbb{R}^{nr_\ell \times nr_\ell}$, and the vectors $\tilde{\mu}^{(w)} = ({\mu_1^{(w)}}^\top \cdots {\mu_n^{(w)}}^\top)^\top \in \mathbb{R}^{nr}$, $(\tilde{\mu}^\ell)^{(w)} = \big((\mu_1^\ell)^{(w)\top} \cdots (\mu_n^\ell)^{(w)\top}\big)^\top \in \mathbb{R}^{nr_\ell}$, $y^\ell = ({y_1^\ell}^\top \cdots {y_n^\ell}^\top)^\top \in \mathbb{R}^{n_\ell}$ with $n_\ell = \sum_{i=1}^n n_i^\ell$ and $y = (y_1^\top \cdots y_n^\top)^\top \in \mathbb{R}^{\mathcal{N}}$ with $\mathcal{N} = \sum_{i=1}^n n_i$. The β update then rewrites

$$\beta^{(w+1)} = (U^\top U)^{-1} U^\top (y - V\tilde{\mu}^{(w)}).$$

For the D update, one has

$$D^{(w+1)} = n^{-1}\big(\text{sum}(\Omega^{(w)}) + \mu^{(w)} {\mu^{(w)}}^\top\big).$$

And finally for the ϕ update, one has

$$\begin{aligned} \phi_\ell^{(w+1)} = n_\ell^{-1}\big[&(y^\ell - U^\ell \beta_\ell^{(w+1)})^\top \big(y^\ell - U^\ell \beta_\ell^{(w+1)} - 2V^\ell (\tilde{\mu}^\ell)^{(w)}\big) \\ &+ \text{Tr}\big\{ {V^\ell}^\top V^\ell \big(\Omega^{\ell(w)} + (\tilde{\mu}^\ell)^{(w)} {(\tilde{\mu}^\ell)^{(w)}}^\top\big)\big\}\big]. \end{aligned}$$

In our implementation, these parameters are initialized with univariates fits using the function `mixedlm` (linear mixed effects model) in the `Python` package `statsmodels`.

A.3.2 Details of the simulation setting

We assume that each of the n subjects belongs to one of two different profiles: high-risk and low-risk. Let us denote by $\mathcal{H} \subset \{1, \ldots, n\}$ the set of high-risk subjects. For generating the time-independent features matrix of subject i, we take

$$X_i \in \mathbb{R}^p \sim \begin{cases} \mathcal{N}\big(\mu, \Sigma_1(\rho_1)\big) & \text{if } i \notin \mathcal{H}, \\ \mathcal{N}\big(-\mu, \Sigma_1(\rho_1)\big) & \text{if } i \in \mathcal{H}, \end{cases}$$

where the mean μ corresponds to the gap between time-independent features of high-risk subjects and low-risk subjects, and $\Sigma_1(\rho_1)$ a $p \times p$ Toeplitz covariance matrix (Mukherjee and Maiti, 1988) with correlation $\rho_1 \in (0,1)$, that is, $\Sigma_1(\rho_1)_{jj'} = \rho_1^{|j-j'|}$. In order to simulate the class g_i for each subject i, we choose a sparse coefficient vector where we decide to keep only $\bar{p}$ active features, that is

$$\xi = (\nu, \ldots, \nu, 0, \ldots, 0) \in \mathbb{R}^p, \tag{A.21}$$

with $\nu \in \mathbb{R}$ being the value of the active coefficients. Then, we generate $g_i \sim \mathcal{B}\big(\pi_\xi(X_i)\big)$, where $\mathcal{B}(\alpha)$ denotes the Bernoulli distribution with parameter $\alpha \in [0,1]$ and

$$\pi_\xi(X_i) = \frac{e^{X_i^\top \xi}}{1 + e^{X_i^\top \xi}}$$

Now, concerning the simulation of longitudinal markers, the idea is to sample from multivariate normal distributions. Moreover, we want to induce sparsity in the longitudinal data to reduce correlation between longitudinal features in each class k. We denote by $\mathcal{S}_k$ the set of active longitudinal features in class k, which is randomly selected from the set $\{1, \ldots, L\}$. Then, we simulate longitudinal features of the form

$$Y_i^\ell(t) = \sum_{k=1}^{K} \mathbb{1}_{\{g_i = k\}} \Big(\big((1,t)^\top \beta_k^\ell + (1,t)^\top b_i^\ell\big) \mathbb{1}_{\{\ell \in \mathcal{S}_k\}} + \varepsilon_i^\ell(t) \Big)$$

where $t \geq 0$, the error term $\varepsilon_i^\ell(t) \sim \mathcal{N}(0, \sigma_\ell^2)$, the global variance-covariance matrix for the random effects components is such that $D = \Sigma_2(\rho_2)$, a $r \times r$ Toeplitz covariance matrix with correlation $\rho_2 \in (0,1)$, and the fixed effect parameters are generated according to

$$\beta_k^\ell \sim \mathcal{N}\Big(\mu_k, \begin{bmatrix} \rho_3 & 0 \\ 0 & \rho_3 \end{bmatrix}\Big)$$

for $k \in \{1,2\}$ and with correlation $\rho_3 \in (0,1)$. The number of observations for each subject is randomly selected from 1 to 10, and the measurement times are simulated from a uniform distribution with mininmum zero and maximum its survival time.

Now to generate survival times, we choose a risk model with

$$\lambda_i(t \,|\, g_i = k) = \lambda_0(t) \exp\Big(\sum_{\ell=1}^{L} \Psi_{i,k}^\ell(t) \gamma_k^\ell \Big), \tag{A.22}$$

We choose a Gompertz distribution (Gompertz, 1825) for the baseline, that is

$$\lambda_0(t) = \kappa_1 \kappa_2 \exp(\kappa_2 t) \tag{A.23}$$

with $\kappa_1 > 0$ and $\kappa_2 \in \mathbb{R}$ the scale and shape parameters respectively, which is a common distribution choice in survival analysis (Klein and Moeschberger, 2005) with a rich history in describing mortality curves. For the choice of the association features, we consider the

two functionals in form of random effects linear predictor $m_k^\ell(t)$ (Chi and Ibrahim, 2006) and random effects b^ℓ (Hatfield et al., 2011), that is $\Psi_{i,k}^\ell(t) = \left(\beta_{k,1}^\ell + \beta_{k,2}^\ell t + b_{i,1}^\ell + b_{i,2}^\ell t \, , b_i^{\ell\top}\right)^\top$ and $\gamma_k^\ell = \nu_k \mathbb{1}_{\{\ell \in \mathcal{S}_k\}}$ where $\nu_k \in \mathbb{R}$ is the coefficients of active association features for each group k. Then one can write

$$\lambda_i(t|g_i = k) = \lambda_0(t) \exp(\iota_{i,k,1} + \iota_{i,k,2} t),$$

being a Cox model with a linear relationship between time-varying feature and log hazard that allows the following explicit survival times generation process. One can now generate survival times explicitly, via the inversion method, as

$$T_i^\star | g_i = k \sim \frac{1}{\iota_{i,k,2} + \kappa_2} \log\left(1 - \frac{(\iota_{i,k,2} + \kappa_2) \log U_i}{\kappa_1 \kappa_2 \exp \iota_{i,k,1}}\right) \tag{A.24}$$

where $U_i \sim \mathcal{U}\big([0,1]\big)$, see Austin, 2013. The distribution of the censoring variable C_i is the geometric distribution $\mathcal{G}(\alpha_c)$, where $\alpha_c \in (0,1)$ is empirically tuned to maintain a desired censoring rate $r_c \in [0,1]$. The choice of all hyper-parameters is driven by the applications on simulated data presented in Section 3.5.1 in the main paper, and summarized in Table A.2.

Table A.2 – Hyper-parameter choices for simulation with $n = 500$, $L = 5$ and $p = 10$.

$\lvert\mathcal{H}\rvert$	$\lvert\mathcal{S}_k\rvert$	(ρ_1, ρ_2, ρ_3)	μ	μ_1	μ_2	σ_ℓ^2	(κ_1, κ_2)	(ν, ν_1, ν_2)	r_c	$\bar{p}$
200	2	$(0.5, 0.01, 0.01)$	1	$\begin{pmatrix} -0.6 \\ 0.1 \end{pmatrix}$	$\begin{pmatrix} 0.05 \\ 0.2 \end{pmatrix}$	0.25	$(0.05, 0.1)$	(0.2, 0.1, 0.4)	0.3	5

A.3.3 Description of the datasets used in comparison study

JoineRML simulation We use the classical R package `joineRML` (Hickey et al., 2018) to simulate multivariate longitudinal and time-to-event data from a joint model. The multivariate longitudinal features are generated for all possible measurement times using multivariate Gaussian linear mixed model. Failure times are simulated from proportional hazards time-to-event models. We sample two time-independent features and two longitudinal features for 250 individuals.

PBCseq dataset This dataset which is available in the R package `JMbayes` (Rizopoulos, 2016a), contains the follow-up of 312 patients with primary biliary cirrhosis, a rare autoimmune liver disease. Several longitudinal features are measured over time (for example serum bilirubin, serum cholesterol, albumin), along with information on gender, age, and drug used recorded once at the beginning of the study. Time-to-event is also recorded with a censoring rate of 55%.

Aids dataset This dataset which is available in the R package `JMbayes` (Rizopoulos, 2016a), compares the efficacy and safety of two drugs for 467 patients diagnosed with HIV

who were either intolerant or resistant to zidovudine therapy. Information on gender, age, drug used, AIDS infection status, and level of intolerance to zidovudine is collected at the start of the study. The longitudinal feature of interest here is the measurement of the number of CD4 cell (a type of white blood cell), a laboratory test used to understand the progression of HIV disease. Time-to-event is also recorded with a censoring rate of 40%.

A.3.4 Procedure to evaluate model performance

Let us describe now in Algorithm 2 the procedure we follow to evaluate model performance on simulated data and real data described in Section 3.5 in the main paper.

Algorithm 2 Procedure followed to assess performances of a given model in our real-time prediction paradigm.

Input: Dataset $\mathcal{D}_n$; a `model` under study
Output: Confidence intervals on C-index metric as well as on running time.

We run $K^{\text{iter}} = 50$ independent experiments
for $k = 1, \ldots, K^{\text{iter}}$ **do**
  $\texttt{start\_time} = \texttt{time}()$
  $(\mathcal{D}_{n_{\text{train}}}, \mathcal{D}_{n_{\text{test}}}) \leftarrow \texttt{split\_train\_test}(\mathcal{D}_n)$
  $\texttt{model.fit}(\mathcal{D}_{n_{\text{train}}})$
  for $i = 1, \ldots, n_{\text{test}}$ **do**
    $s_i \sim \max_{\ell \in \{1,\ldots,L\}} (t^{\ell}_{in_i}) \times \big(1 - \text{Beta}(2,5)\big)$
    $Y_i \leftarrow \big(Y_i^{\ell}(t_{ij}^{\ell})\big)_{\substack{j \in \{1,\ldots,n_i^{\ell}-1\} \\ \ell \in \{1,\ldots,L\}}}$ with $t_{ij}^{\ell} \leq s_i$
    $\mathcal{X}_i = (X_i, Y_i)$
    $\widehat{\mathcal{R}}_i^k(s_i) \leftarrow \texttt{model.predict}(\mathcal{X}_i)$
  end for
  $\texttt{score}_k \leftarrow \texttt{c\_index}\big((\widehat{\mathcal{R}}_i^k(t_i), T_i, \Delta_i)_{i=1,\ldots,n_{test}}\big)$
  $\texttt{end\_time} = \texttt{time}()$
  $\texttt{running\_time}_k = \texttt{end\_time} - \texttt{start\_time}$
end for
Return $\hat{\theta} = \theta^{(w+1)}$

Screening phase. We use the multivariate Cox model and C-index metric for selecting the M most important features from a specific set of feature extraction functions $\mathcal{F}$. For each feature in $\mathcal{F}$, we extract the $n \times L$ matrix from all L longitudinal markers of all n subjects. Then we fit this extracted matrix with all the survival times T and censoring indicators Δ in the Cox model and use the C-index metric to evaluate the performance. Finally, we select M features have the highest C-index values.

A.3.5 Interpretation of the model on medical datasets

Tables A.3a and A.3b provide the estimated coefficients of FLASH on the PBCseq and Sepsis datasets. Coefficients are organized as follows: first, the time-independent parame-

ters ξ, then the coefficients corresponding to the association features of the low-risk group $\gamma^{\ell}_{1,m}$ and finally the ones of the high-risk group $\gamma^{\ell}_{2,m}$. The initial values of the longitudinal markers are also considered as time-independent features in these experiments. Since each longitudinal marker is associated to a vector of association features, what we call "coefficient" is actually the Euclidean norm of the coefficients associated to these association features, that is, the norm of $(\gamma^{\ell}_{k,1}, \ldots, \gamma^{\ell}_{k,M})$. In addition, to obtain standard errors for the coefficient estimates, we use a Bootstrap approach adapted to the presence of a Lasso penalty following Chzhen et al. (2019). We first run the model with the Lasso penalty to get the support of estimated coefficients. We then rerun the model 10 times without the Lasso penalty on bootstrap samples with only features whose coefficients are in the support of the first run.

Note that with this approach the errors of estimation of the coefficients that are not in the support of the first run are then not evaluated. In other words, we do not evaluate the stability of our variable selection approach. To do this, we could follow the procedure of Bach (2008) who suggest running several runs of Lasso on a bootstrap sample, and then looking at the different support of the coefficients.

A.3.6 Experiments on a high-dimensional dataset

We evaluate the performance of FLASH model with a challenging high-dimensional dataset from NASA, which is available at `https://data.nasa.gov`. This dataset describes the degradation of 200 aircraft engines, where 17 multivariate longitudinal features are measured for each different aircraft engine until its failure. There are also three operational settings that significantly affect engine performance. Note that we only apply FLASH to this dataset because the other models did not converge after running for one day, highlighting the fact that they do not scale to high-dimensional settings.

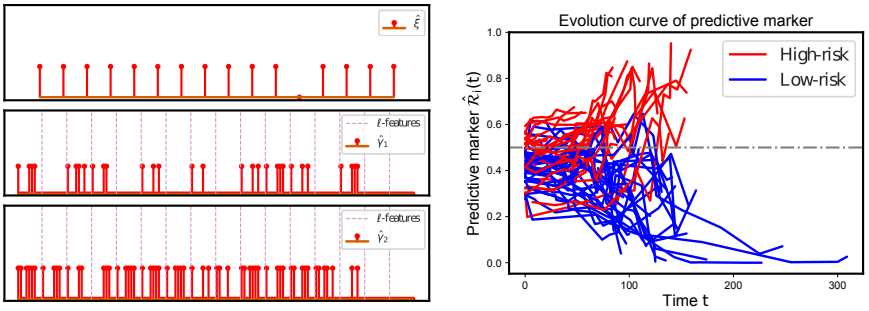

Figure A.1 – NASA dataset results. Left: in red the support of the estimated coefficient $\hat{\xi}$ ab $\hat{\gamma}_k$ for $k \in \{1, 2\}$, the dashed pink lines separate the features corresponding to each longitudinal marker ℓ. Right: the evolution curves of the predictive marker $\widehat{\mathcal{R}}_i(t)$ for varying times t and each subject where we well separate the subjects in the high risk group (in red color) and low risk (in blue color) based on their predictive marker at last measurement time with a threshold 0.5 represent by horizontal dashed line.

We illustrate in Figure A.1 the results obtained by FLASH. In the left panel, we can see the effect of regularization where the coefficients learned by the model are sparse and

some longitudinal markers are entirely discarded. In particular, five longitudinal markers are excluded for the first group $k = 1$ but not for the group $k = 2$ while the last two markers are never selected. In the right panel, we show the evolution in time of the predictive marker for each subject. We can see that, as time passes, more data is observed and the subjects are better separated into two groups of different risks.

A.3.7 Experiments on using signatures as association functions

Signature transform The association features can be extracted by a signature transformation. We refer to Fermanian (2021) for a detailed presentation of this transformation and simply recall here its definition. Let $I = (\ell_1, \dots, \ell_k)$ be a word of size k from the alphabet $\{1, \dots, L\}^k$. The signature associated to I is defined as the mapping

$$t \mapsto \mathbf{S}^I(\mathcal{Y}_i(t^-)) := \int_{0<u_1<\cdots<u_k<t} dy_i^{\ell_1}(u_1) \dots dy_i^{\ell_k}(u_k).$$

The signature of depth N is defined as the vector

$$\mathbf{S}_N(\mathcal{Y}_i(t^-)) = \left(\mathbf{S}^I(\mathcal{Y}_i(t^-))\right)_{|I|\leq N}.$$

It is a transformation from a multivariate longitudinal marker to a sequence of coefficients, that are independent of time parameterization and encodes geometric properties (for example, the second order coefficients correspond to areas). It is therefore a very different transformation from the ones used in the `tsfresh` package, in particular because it encodes information on interactions between coordinates, whereas `tsfresh` focuses on univariate features. The truncation depth N is an hyperparameter that can be selected typically by cross-validation.

Results Figure A.2 shows the performance of FLASH with signatures as association features compared with the performance of the model with the feature extracted from *tsfresh* and the two competing methods on the four datasets presented in the article. The prediction performance of the model with the signature transformation is comparable to the competing methods and better for the FLASH_simu dataset, and its computation cost is reduced since it does not require to implement a screening phase procedure to select the top association features.

A.3.8 Procedure to select the optimal number of latent groups

The optimal number of latent groups, K, is selected based on the Bayesian information criterion (BIC) (Hastie et al., 2009), which is defined as

$$\text{BIC}(K) = -2\hat{\mathcal{L}}_n^{\text{pen}}(\theta; K) + \log(n)K,$$

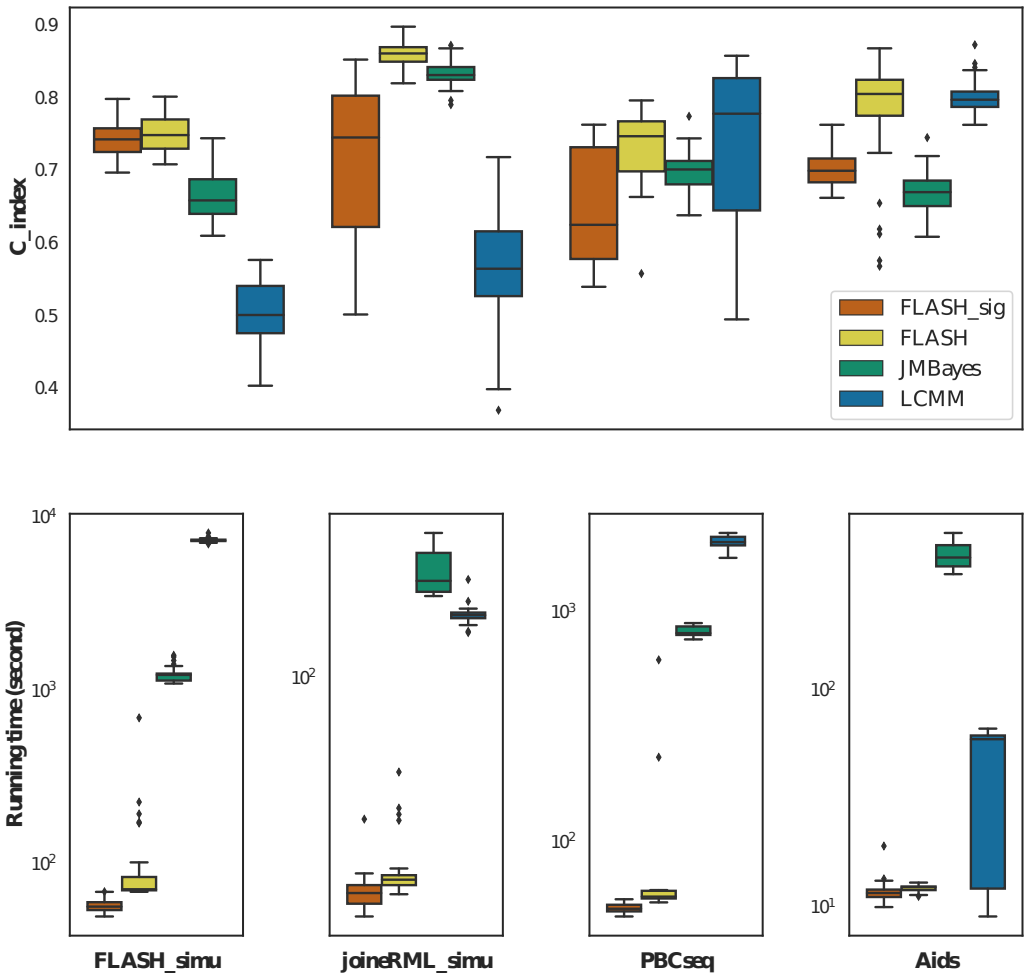

Figure A.2 – The performance of the Flash model with the signature transform compares with the competing methods.

where $\hat{\mathcal{L}}_n^{\text{pen}}(\theta; K)$ is the optimal value of the likelihood function (defined in (3.8) the main paper) with K groups and n is the number of subjects. The optimal K is selected with the "elbow method", that is, we pick the value of K corresponding to the first large drop in the BIC values. For example, Figure A.3 shows the curve of the BIC values obtained with different K on the PBCseq dataset. In this case, the optimal K is 4.

A.3.9 Sensitivity to latent class assumptions

To assess the sensitivity of the method, we have run an additional simulation where we deviate from the assumption that the latent class is generated from (3.1) in the main paper. More precisely, we have used a probit model (Aldrich and Nelson, 1984), where the probability of belonging to a class has the form:

$$\mathbb{P}(g_i = k) = \Phi(X_i^\top \xi_k),$$

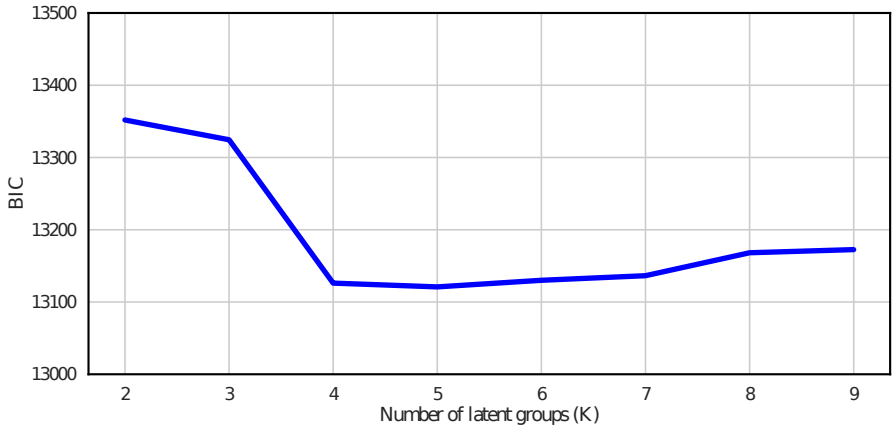

Figure A.3 – The BIC values on the PBCseq dataset.

where Φ is the cumulative distribution function of the standard normal distribution. In Figure A.4 below, we compare the performance of our method on the dataset simulated in the "well-specified" setting, described in Section 5.1 of the paper, and on a dataset simulated in this misspecified setting. We can see that the performance is slightly better in the Lo-

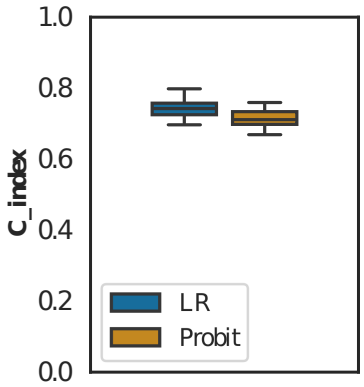

Figure A.4 – The performance of FLASH in terms of C-index, on two simulated datasets: the LR (Logistic Regression) dataset is the one simulated with the model (3.1) in the main paper and the Probit dataset described above.

gistic Regression case, which was to be expected, but that this difference is not significant. Therefore, our proposed method is not too sensitive to the assumption of the model on the probability of latent class membership.

Table A.3 – Estimated coefficients with standard errors.

(a) PBCseq dataset

Features	Coefficient
Drug	0.0
Age	0.0
Sex	0.3459 ± 0.0037
Initial value of Serbilir	0.3476 ± 0.0091
Initial value of Albumin	0.0
Initial value of SGOT	0.3664 ± 0.0069
Initial value of Platelets	0.0
Initial value of Prothrombin	0.3587 ± 0.0088
Initial value of Alkaline	0.332 ± 0.0042
Initial value of SerChol	0.2978 ± 0.0093
Serbilir	0.0341 ± 0.0161
Albumin	0.0833 ± 0.0088
SGOT	0.0
Platelets	0.0325 ± 0.0159
Prothrombin	0.0819 ± 0.0129
Alkaline	0.0807 ± 0.0092
SerChol	0.0
Serbilir	0.0
Albumin	0.0
SGOT	0.0
Platelets	0.0
Prothrombin	0.0
Alkaline	0.2788 ± 0.0355
SerChol	0.0

(b) Sepsis dataset

Features	Coefficient
Age	0.152 ± 0.0414
Gender	0.0
Temp	0.0
DBP	0.0669 ± 0.1234
BaseExcess	0.2779 ± 0.0404
HCO3	0.0
pH	0.0
PaCO2	0.119 ± 0.0987
BUN	0.0
Calcium	0.0
Chloride	0.0
Creatinine	0.0
Glucose	0.0
Magnesium	0.0
Phosphate	0.0
Potassium	0.0
Hct	0.0
Hgb	0.0
PTT	0.0
WBC	0.0
Platelets	0.0
Initial value of HR	0.0
Initial value of O2Sat	0.1325 ± 0.1049
Initial value of SBP	0.6485 ± 0.0788
Initial value of Resp	0.7821 ± 0.1001
HR	0.0
02Sat	0.0
SBP	0.1111 ± 0.0211
Resp	0.1016 ± 0.0176
HR	0.0198 ± 0.0051
02Sat	0.0783 ± 0.0066
SBP	0.0849 ± 0.0064
Resp	0.0809 ± 0.0056

Appendix B

Supplementary material of Chapter 4

Contents

B.1 Supplementary Mathematical Elements

B.1.1 Supplementary elements on survival analysis

The counting process associated with the observation of $T_1^i < T_2^i < \dots$ is denoted by $\tilde{N}^i$. The observed counting process is $t \to N^i(t) = \int_0^t Y^i(s) d\tilde{N}^i(s)$. The integral against the counting process N^i is to be understood is to be understood as a Stieltjes integral, i.e., $\int_0^t \lambda_\star^i(s) dN^i(s) = \sum_{T_i \leq t} \lambda_\star^i(T_i)$ — see Aalen et al. (2008, p.55-56). Its intensity writes $\lambda_\star^i(t \,|\, \mathbf{W}^i, (x^i(s))_{s \leq t}) Y^i(t)$, which we simply write $\lambda_\star^i(t) Y^i(t)$ to alleviate notations.

To the observations, we associate the filtration $\mathcal{F}$, with all σ-fields at $0 \leq t \leq \tau$ defined as

$$\mathcal{F}_t = \bigcup_{i=1,\dots,n} \mathcal{F}_t^i$$

where $\mathcal{F}_t^i = \sigma\big(x^i(s), \mathbf{W}^i, N^i(s), Y^i(s), 0 \leq s \leq t\big)$. We assume in addition that Y^i is $\mathcal{F}^i$-predictable.

Using the Doob-Meyer decomposition of counting processes - see Aalen et al. (2008, p. 52-60) - we have

$$N^i(t) = \int_0^t \lambda_\star^i(s) Y^i(s) ds + M^i(t) \tag{B.1}$$

where M^i is local square integrable martingale with respect to $\mathcal{F}^i$.

B.1.2 Picard-Lindelhöf Theorem

Theorem 2. Let $x : [0, \tau] \to \mathbb{R}^d$ be a continuous path of bounded variation, and assume that $\mathbf{G} : \mathbb{R}^p \to \mathbb{R}^{p \times d}$ is Lipschitz continuous. Then the CDE

$$dz(t) = \mathbf{G}(z(t)) dx(t)$$

with initial condition $z_0 \in \mathbb{R}^p$ has a unique solution on $[0, \tau]$.

A full proof can be found in Fermanian et al. (2021, Theorem 4). Remark that in our setting, NCDEs are Lipschitz since typical neural vector fields, such as feed-forward neural networks, are Lipschitz (Virmaux and Scaman, 2018). This ensures that the solutions to NCDEs are well-defined.

B.1.3 Continuity of the Flow of CDEs

We state a result on the continuity of the flow adapted from Bleistein and Guilloux (2024), Theorem B.5.

Theorem 3. Let $\mathbf{F}, \mathbf{G} : \mathbb{R}^p \to \mathbb{R}^{p\times d}$ be two Lipschitz vector fields with Lipschitz constants $L_{\mathbf{F}}, L_{\mathbf{G}} > 0$. Let $x, r : [0,\tau] \to \mathbb{R}^d$ be either continuous or piecewise constant paths of total variations bounded by L_x and L_r. Consider the controlled differential equations

$$dw(t) = \mathbf{F}(w(t))dx(t) \text{ and } dv(t) = \mathbf{G}(v(t))dr(t)$$

with initial conditions $w(0) = v(0) = 0$ respectively. It then follows that for any $t \in [0,\tau]$

$$\|w(t) - v(t)\| \leq \left(\|x - r\|_{\infty,[0,t]} \left(1 + L_{\mathbf{F}} L_r \mathcal{K}\right) + \max_{v\in\Omega} \|\mathbf{F}(v) - \mathbf{G}(v)\|_{\text{op}} L_r \right) \exp(L_{\mathbf{F}} L_x),$$

where

$$\mathcal{K} = \left[L_{\mathbf{F}} \left(\|\mathbf{F}(0)\|_{\text{op}} L_x \right) \exp(L_{\mathbf{F}} L_x) + \|\mathbf{F}(0)\|_{\text{op}} \right] \exp(L_{\mathbf{F}} L_x)$$

and

$$\Omega = \left\{ u \in \mathbb{R}^p \mid \|u\| \leq (\|\mathbf{G}(0)\|_{\text{op}} L_r) \exp(L_{\mathbf{G}} L_r) \right\}.$$

B.1.4 Linearization in the Signature Space

General Result

In this section, we give additional details on the linearization of CDEs in the signature space. We first define the differential product.

Definition B.1.1. Let $F, G : \mathbb{R}^p \to \mathbb{R}^p$ be two $\mathcal{C}^\infty$ vector fields and let $J(\cdot)$ be the Jacobian matrix. Their differential product $F \star G : \mathbb{R}^p \to \mathbb{R}^p$ is the smooth vector field defined for every $h \in \mathbb{R}^p$ by

$$(F \star G)(h) = \sum_{j=1}^{e} \frac{\partial G}{\partial h_j}(h) F_j(h) = J(G)(h) F(h).$$

We now consider a tensor field $\mathbf{F} : \mathbb{R}^p \to \mathbb{R}^{p\times d}$ which we write

$$\mathbf{F} = \begin{bmatrix} | & \dots & | \\ F^1 & \dots & F^d \\ | & \dots & | \end{bmatrix},$$

where for every $1 \leq i \leq d$, $F^i : \mathbb{R}^p \to \mathbb{R}^p$, and we define

$$\Gamma_k(\mathbf{F}) := \sup_{\|h\|\leq M,\, i_1 \leq \dots \leq i_k \leq d} \left\| F^{i_1} \star \dots \star F^{i_k}(h) \right\|_2.$$

Consider the solution $z : [0, \tau] \to \mathbb{R}^p$ to the CDE

$$\begin{aligned} dz(t) &= \mathbf{F}(z(t))dx(t) \\ z(0) &= \mathbf{0} \in \mathbb{R}^p \end{aligned} \tag{B.2}$$

where $x : [0, \tau] \to \mathbb{R}^d$ is a continuous path of finite total variation bounded by $L_x\tau > 0$. We recall the following result from Fermanian et al. (2021), Proposition 4.

Proposition 4 (Fermanian et al. (2021), Proposition 4.)**.** We have

$$\left\| z_\star^i(t) - \alpha_{\star,N}^\top \mathbf{S}_N(x_{[0,t]}) \right\| \leq \frac{(dL_x t)^{N+1}}{(N+1)!} \Gamma_{N+1}(\mathbf{F})$$

As a consequence, we have the following theorem.

Theorem 5. Let $\mathbf{F} : \mathbb{R}^p \to \mathbb{R}^{p \times d}$ be a $\mathcal{C}^\infty$ tensor field. If

$$\frac{(dL_x t)^{N+1}}{(N+1)!} \Gamma_{N+1}(\mathbf{F}) \to 0$$

as $N \to +\infty$, then the solution z to the CDE (B.2) can be written as

$$z(t) = \sum_{k \geq 1} \sum_{I \in \{1,\dots,d\}^k} \mathbf{S}^I(x_{[0,t]}) F^{i_1} \star \cdots \star F^{i_k}(\mathbf{0}).$$

Application to our Model

Recall that we have defined our generative model through the CDE

$$dz_\star^i(t) = \mathbf{G}_\star(z_\star^i(t))dx^i(t)$$

with initial condition $z_\star^i(0) = 0$, where $\mathbf{G}_\star : \mathbb{R} \to \mathbb{R}^p$ is a $L_{\mathbf{G}_\star}$-Lipschitz vector field. Since in our case, the vector field $\mathbf{G}_\star$ maps $\mathbb{R}$ to $\mathbb{R}^d$, it can be written as

$$\mathbf{G}_\star(z) = \begin{bmatrix} G_\star^1(z) & \dots & G_\star^d(z), \end{bmatrix},$$

where for every $1 \leq i \leq d$, $G_\star^i : \mathbb{R} \to \mathbb{R}$. In this setup, for $1 \leq i_1, i_2 \leq d$ the differential product collapses to

$$(G_\star^{i_1} \star G_\star^{i_2})(h) = (G_\star^{i_2})'(h) \times G_\star^{i_1}(h) \in \mathbb{R}.$$

For $1 \leq i_1, i_2, i_3 \leq d$, it writes

$$\begin{aligned} (G_\star^{i_1} \star G_\star^{i_2} \star G_\star^{i_3})(h) &= (G_\star^{i_2} \star G_\star^{i_3})'(h) \times G_\star^{i_1}(h) \\ &= \left((G_\star^{i_3})'(h) \times G_\star^{i_2}(h)\right)' \times G_\star^{i_1}(h) \\ &= \left((G_\star^{i_3})^{(2)}(h) \times G_\star^{i_2}(h) + (G_\star^{i_3})'(h) \times (G_\star^{i_2})'(h)\right) \times G_\star^{i_1}(h) \in \mathbb{R}. \end{aligned}$$

One can derive the similar expression for $1 \leq i_1, \dots, i_k \leq d$. In line with Theorem 5, we make the following Assumption on the vector field $\mathbf{G}_\star$.

Assumption 7. The vector field $\mathbf{G}_\star$ satisfies

$$\frac{(L_x \tau d)^{N+1}}{(N+1)!} \Gamma_{N+1}(\mathbf{G}_\star) \to 0$$

as $N \to \infty$.

We can write the ℓ_2 and ℓ_1 norms of $\alpha_{\star,N}$ as functions of the differential product of $\mathbf{G}_\star$.

Lemma 9. We have that

$$\|\alpha_{\star,N}\|_2 \leq \Big(\sum_{k=1}^{N} d^k \Gamma_k(\mathbf{G}_\star)^2 \Big)^{1/2}$$

and

$$\|\alpha_{\star,N}\|_1 \leq \sum_{k=1}^{N} d^k \Gamma_k(\mathbf{G}_\star).$$

Proof. Starting with the ℓ_2 norm, one has

$$\begin{aligned}
\|\alpha_{\star,N}\|_2 &= \Big(\sum_{k=1}^{N} \sum_{1 \leq i_1, i_2, \dots, i_k \leq d} G_\star^{i_1} \star \cdots \star G_\star^{i_k}(\mathbf{0})^2 \Big)^{1/2} \\
&\leq \Big(\sum_{k=1}^{N} d^k \max_{1 \leq i_1, i_2, \dots, i_k \leq d} |G_\star^{i_1} \star \cdots \star G_\star^{i_k}(\mathbf{0})|^2 \Big)^{1/2} \\
&\leq \Big(\sum_{k=1}^{N} d^k \Gamma_k(\mathbf{G}_\star)^2 \Big)^{1/2}.
\end{aligned}$$

Moving on to the ℓ_1 norm, we similarly obtain

$$\begin{aligned}
\|\alpha_{\star,N}\|_1 &= \Big(\sum_{k=1}^{N} \sum_{1 \leq i_1, i_2, \dots, i_k \leq d} |G_\star^{i_1} \star \cdots \star G_\star^{i_k}(\mathbf{0})| \Big) \\
&\leq \Big(\sum_{k=1}^{N} d^k \max_{1 \leq i_1, i_2, \dots, i_k \leq d} |G_\star^{i_1} \star \cdots \star G_\star^{i_k}(\mathbf{0})| \Big) \\
&\leq \sum_{k=1}^{N} d^k \Gamma_k(\mathbf{G}_\star).
\end{aligned}$$

B.1.5 Signature of a Discretized Path

We recall the following result from Bleistein et al. (2023).

Theorem 6. Let $x : [0, \tau] \to \mathbb{R}^d$ be a path satisfying Assumption 4. Let $D = \{t_1, \dots, t_K\} \subset [0, \tau]$ be a grid of sampling points, and x^D the piecewise constant interpolation of the path x sampled on the grid D. For all $\alpha \in \mathbb{R}^q$, where $q := \frac{d^N-1}{d-1}$, we have

$$|\alpha^\top \big(\mathbf{S}_N(x_{[0,t]}) - \mathbf{S}_N(x^D_{[0,t]})\big)| \leq c_3(N)\, \|\alpha\|\, |D|,$$

where

$$c_3(N) = 2e\frac{(L_x t)^{N-1} - 1}{L_x t - 1} L_x.$$

B.1.6 The Cox Connection

Signature-based embeddings. Consider a continuous path of bounded variation $x : t \mapsto (x(t), t) \in \mathbb{R}^d$. First, remark that for every word of size one $I \in \{1, \dots, d\}$, the signature writes

$$\mathbf{S}^I(x_{[0,t]}) = \int_{0<u_1<t} dx^{(I)}(s) = x^{(I)}(t).$$

Furthermore, for any word $I = (d, \dots, d)$ of size k made only of the letter d, i.e., words that only include the time channel, we have

$$\mathbf{S}^I(x_{[0,t]}) = \int_{0<u_1<\cdots<u_k<t} du_1 \dots du_k = \frac{1}{k!} t^k.$$

This shows that for $x = x^{i,D}$

$$\alpha^\top \mathbf{S}_N(x^{i,D}_{[0,t]}) = \alpha_{I_1}^\top (1, t, t^2, \dots, t^N) + \alpha_{I_2}^\top \mathbf{X}^i(t) + \sum_{I \in I_3} \alpha_I \mathbf{S}^I(x^{i,D}_{[0,t]})$$

where α_{I_1} is the subvector of α collecting all coefficients associated to the words $\{d\}, \{d, d\}, \dots, \{d, \dots, d\}$ containing only the letter d, α_{I_2} is the subvector collecting all coefficients associated to the $d-1$ words $\{1\}, \{2\}, \dots, \{d-1\}$ of size 1, and α_{I_3} collects the remaining coefficients.

NCDEs. For any $N \geq 1$, consider the augmented vector field

$$\tilde{\mathbf{G}}_\psi(z) = \begin{bmatrix} \mathbf{G}_\psi(z) & \mathbf{0}_{p\times(N-1)} \\ \mathbf{0}_{(N-1)\times d} & \mathbf{I}_{(N-1)\times(N-1)} \end{bmatrix} \in \mathbb{R}^{(N-1+p)\times(N-1+d)}, \quad z \in \mathbb{R}^p,$$

and an embedding of the time series $\mathbf{X}^i$ of the form $\tilde{x}^{i,D}(s) = (\mathbf{X}^i(t_k), s, s^2, \dots, s^N) \in$

$\mathbb{R}^{d+N-1}$ for $s \in [t_k, t_{k+1}[$. The latent state of the NCDE model is now updated as

$$\begin{aligned}\tilde{z}_\theta^{i,D}(t_{k+1}) &= \tilde{z}_\theta^{i,D}(t_k) + \tilde{\mathbf{G}}_\psi(\tilde{z}_\theta^{i,D}(t_k))\Delta\tilde{\mathbf{X}}^i(t_{k+1}) \\ &= \tilde{z}_\theta^{i,D}(t_k) + \begin{bmatrix} \mathbf{G}_\psi(z_\theta^{i,D})\Delta\mathbf{X}^i(t_{k+1}) \\ \Delta t_{k+1} \\ \vdots \\ \Delta t_{k+1}^N \end{bmatrix} = \begin{bmatrix} z_\theta^{i,D}(t_k) + \mathbf{G}_\psi(z_\theta^{i,D})\Delta\mathbf{X}^i(t_{k+1}) \\ t_{k+1} \\ \vdots \\ t_{k+1}^N \end{bmatrix}.\end{aligned}$$

This proves that in the NCDE-based model, the intensity can similarly be written as

$$\alpha^\top \tilde{z}_\theta^{i,D}(t) = \alpha_{I_1}^\top z_\theta^{i,D}(t) + \alpha_{I_2}^\top (1, t, t^2, \dots, t^N)$$

where $\alpha = (\alpha_{I_1}, \alpha_{I_2})$, and $\alpha_1 \in \mathbb{R}^p$ and $\alpha_2 \in \mathbb{R}^N$.

B.1.7 Self-concordance

We now state a self-concordance bound, which can be found along with its proof in Bach (2010).

Lemma 10. Let $g : \mathbb{R} \to \mathbb{R}$ be a convex, three times differentiable function such that

$$|g^{(3)}(x)| \leq M g^{(2)}(x)$$

for all $x \in \mathbb{R}$ and for some $M \geq 0$. Then it follows that

$$\frac{g^{(2)}(0)}{M^2}\Phi(-Mt) \leq g(t) - g(0) - tg'(0) \leq \frac{g^{(2)}(0)}{M^2}\Phi(Mt)$$

for all $t \geq 0$, where

$$\Phi : t \mapsto \exp(t) - t - 1.$$

B.1.8 Decomposition of the difference in likelihoods

We first define the empirical KL-divergence between the true and parameterized intensity associated to the sample $\mathcal{D}_n$ as

$$\mathrm{KL}_n(\lambda_\star, \lambda_\theta^D) := \frac{1}{n}\sum_{i=1}^n \int_0^\tau \log\frac{\lambda_\star^i(s)}{\lambda_\theta^i(s)}\lambda_\star^i(s)Y^i(s)ds - \frac{1}{n}\sum_{i=1}^n \int_0^\tau \big(\lambda_\star^i(s) - \lambda_\theta^i(s)\big)Y^i(s)ds.$$

This definition is classical for intensities of counting processes (Aalen et al., 2008; Lemler, 2016). We now show that minimizing the empirical KL-divergence between the true and the parameterized intensity amounts to minimizing the empirical log-likelihood, ignoring a noise term that will be canceled by setting the penalty accordingly.

Proposition 7. For every $\theta \in \Theta$, the difference in likelihoods $\ell_n^D(\theta) - \ell_n^\star$ decomposes as

$$\mathrm{KL}_n(\lambda_\star, \lambda_\theta^D) - \frac{1}{n}\sum_{i=1}^{n} \int \log \frac{\lambda_\theta^{i,D}(s)}{\lambda_\star^i(s)} dM^i(s),$$

where $M^i : [0, \tau] \to \mathbb{R}$ is a local square integrable martingale.

This proposition is a consequence of the Doob-Meyer decomposition $N^i(t) = \int_0^t \lambda_\star^i(s) Y^i(s) ds + M^i(t)$ of the counting process (Aalen et al., 2008). We now furthermore define the total variation divergence as

$$\mathrm{TV}_n(\lambda_\star, \lambda_\theta^D) := \frac{1}{n}\sum_{i=1}^{n} \int_0^\tau |\lambda_\star^i(s) - \lambda_\theta^{i,D}(s)| Y^i(s) ds$$

and the quadratic log divergence $\mathrm{D}_n^2(\lambda_\star, \lambda_\theta^D)$ as

$$\frac{1}{n}\sum_{i=1}^{n} \int_0^\tau \big(\log \lambda_\theta^{i,D}(s) - \log \lambda_\star^i(s)\big)^2 \lambda_\star^i(s) Y^i(s) ds.$$

Proposition 8. There exist two constants $c_1, c_2 > 0$ such that

$$c_1 \mathrm{TV}_n(\lambda_\star, \lambda_\theta^D)^2 \leq \mathrm{KL}_n(\lambda_\star, \lambda_\theta^D) \leq c_2 \mathrm{D}_n^2(\lambda_\star, \lambda_\theta^D).$$

More precisely, the constants c_1, c_2 are functions of Θ, L_x, τ and $L_{\mathbf{G}_\star}$ and are given explicitly in Appendix B.2.

This bound is obtained by combining a Pinsker-type inequality and a self-concordance bound (Appendix B.2). It is informative in two ways. First, it shows that minimizing the negative empirical log-likelihood and hence the KL-divergence between the true and parameterized intensity will lead to a minimization of the total variation between the two intensities. Secondly, it shows that the KL-divergence is upper bounded by a term involving the difference of the logarithms of the intensities. We make use of this second bound to obtain a bias-variance decomposition.

B.2 Proofs

We refer to Bleistein et al. (2024) for the detailed proofs.

B.3 Algorithmic and Implementation Details

In this Section, we provide extra information about learning algorithms described in the main paper and their hyperparameters optimization by grid-search.

B.3.1 Description of Competing Methods

CoxSig and CoxSig+

Implementation. We use `iisignature` (Reizenstein and Graham, 2020) to compute signatures. Alternatives for computing signatures include the `signatory` library (Kidger and Lyons, 2020).

Training. We minimize the penalized negative log-likelihood (defined in 4.3 in the main paper) using a vanilla proximal point algorithm (Boyd and Vandenberghe, 2004).

Hyperparameters. The initial learning rate of the proximal gradient algorithm is set to e^{-3} and the learning rate for each iteration is chosen by back tracking linesearch method (Boyd and Vandenberghe, 2004). The hyperparameters of penalization strength (η_1, η_2) and truncation depth N are chosen by 1-fold cross-validation of a mixed metric equal to the difference between the C-index and the Brier score. We select the best hyperparameters that minimize the average of this mixed metric on the validation set. We list the hyperparameters search space of this algorithm below.

- η_1: {1, e^{-1}, e^{-2}, e^{-3}, e^{-4}, e^{-5}};
- η_2: {1, e^{-1}, e^{-2}, e^{-3}, e^{-4}, e^{-5}};
- N: {2, 3}. Larger values were considered in the beginning of experiments but were removed from the cross-validation grid because they yielded bad performance and numerical instabilities.

NCDE

Implementation. We implement the fill-forward discrete update of NCDEs in `Pytorch`.

Structure. The neural vector field is a feed-forward network composed of two fully connected hidden layers whose hidden dimension is set to 128. We choose to represent the latent state in 4 dimensions—the number of nodes in the input layer is therefore set to 4. The dimension of the output layer is equal to the multiplication of the dimension of the hidden layer (128) and the dimension of the sample paths of a given data set. `tanh` is set to be the activation function for all the nodes in the network.

Training. The model was trained for 50 epochs using the Adam optimizer (Kingma and Ba, 2014) with a batch size of 32 and cross-validated learning rate set to e^{-4}.

Cox Model

Implementation and Training. We use a classical Cox model with elastic-net penalty as a baseline, which is given either the first measured value of the individual time series or the static features if they are available. The intensity of this model has then the form

$$\lambda_\theta^i(t) = \lambda_0(t) \exp(\beta^\top \mathbf{W}^i),$$

where $\mathbf{W}^i = \mathbf{X}^i(0)$ if no static features are available. We use the implementation provided in the Python package `scikit-survival` and called `CoxnetSurvivalAnalysis` (Pölsterl, 2020).

Hyperparameters. The ElasticNet mixing parameter γ is set to 0.1. The hyperparameter of penalization strength η is chosen by cross-validation as described above. We crossvalidate over the set $\{1, e^{-1}, e^{-2}, e^{-3}, e^{-4}, e^{-5}\}$ to select the best value.

Random Survival Forest

Implementation. We use the implementation of RSF (Ishwaran et al., 2008) provided in the Python package `scikit-survival` (Pölsterl, 2020).

Training. We train this model with static features $\mathbf{W}^i$ as the only input. Similarly to our implementation of the Cox model, we use the first value of the time series as static features if no other features are available.

Hyperparameters. We cross-validate two hyperparameters on the following grids.

- `max_features`: {None, `sqrt`};
- `min_samples_leaf`: {1, 5, 10};

Dynamic Deep-Hit (Lee et al., 2019)

DDH is a dynamical survival analysis algorithm that frames dynamical survival analysis as a classification problem. It divides the considered time period $[0, \tau]$ into a set of contiguous time intervals. The network is then trained to predict a time interval of event for every subject, which is a multiclass classification task.

Network Architecture. Being adapted to competing events, Dynamic Deep-Hit combines a shared network with a cause-specific network. The *shared network* is a combination of a RNN-like network that processes the longitudinal data and an attention mechanism, which helps the network decide which part of the history of the measurements is important. The *cause-specific network* is a feed-forward network taking as an input the history of embedded measurements and learning a cause-specific representation. See Figure B.1 for a graphical representation of the network's structure.

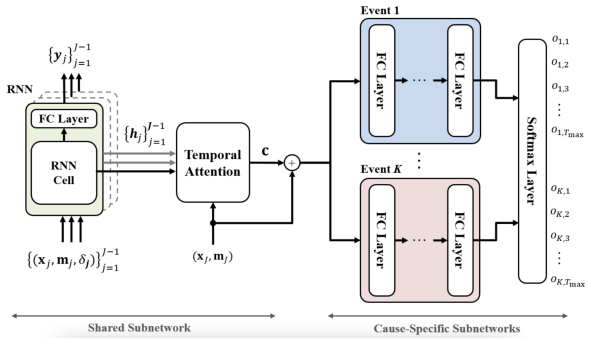

Figure B.1 – Network structure of Dynamic DeepHit. Figure is taken from Lee et al. (2019).

Loss Function. The loss function of DDH is a sum of three loss functions

$$\ell_{\text{Dynamic DeepHit}} = \ell_{\text{log-likelihood}} + \ell_{\text{ranking}} + \ell_{\text{prediction}}.$$

The first loss maximizes the conditional likelihood of dying in the interval $[t_k, t_{k+1}[$ given that the individual has survived up to time t_k. On a side note, we notice that the claim of Lee et al. (2019) that this loss corresponds to "the negative log-likelihood of the joint distribution of the first hitting time and corresponding event considering the right-censoring" of the data is hence inexact. This might explain the results observed in Figure 4.4: DDH's performance, in terms of Brier score, strongly degrades as δt increases because the model is only trained to predict one step ahead, instead of maximizing the full likelihood.

The second loss favors correct rankings among at risk individuals: an individual experiencing an event at time T^i should have a higher risk score at time $t < T^i$ than an individual j for which $T^j > T^i$.

The third loss is a prediction loss, which measures the difference between the value of the time-dependent features and a prediction of this value made by the shared network.

The loss is minimized using Adam (Kingma and Ba, 2014).

Hyperparameters. In our setting, we use the network in its original structure. The learning rate is set to e^{-4} and the number of epochs to 300.

SurvLatent ODE (Moon et al., 2022)

Network Architecture. SurvLatent ODE is a variational autoencoder architecture (Kingma and Welling, 2013). The encoder embeds the entire longitudinal features into an initial latent state, and the decoder uses this latent state to drive the latent trajectory and to estimate the distribution of event time. In this framework, the encoder is an ODE-RNN architecture (Rubanova et al., 2019), which handles the longitudinal features sequentially backward in time and outputs the posterior over the initial latent state. The decoder, which is adapted to competing events, consists of an ODE model and cause-specific decoder modules. The latent trajectory derived from the ODE model is shared across cause-specific decoder modules to estimate the cause-specific discrete hazard functions. See Figure B.2 for a graphical representation of the network's structure.

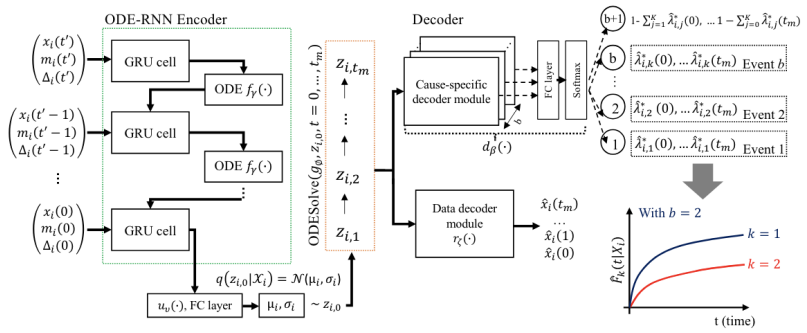

Figure B.2 – Network structure of SurvLatent ODE. Figure taken from Moon et al. (2022).

Loss Function. The loss function is a combination of the log-likelihood and the Kullback-Leibler divergence between the approximate and the true posterior over the initial latent state.

Hyperparameters. In our setting, we use the network in its original structure. The learning rate is set to e^{-2} and the number of epochs to 15, as in the original paper. The training of this framework cannot use subjects whose last longitudinal measurement time is equal to the event time, which is not the case for our proposed methods as well as other competing methods. In order to avoid this problem, we then stop observing the longitudinal measurement before the time-to-event for a period equal to 80 % of the event time of these subjects when training the model of this framework.

B.3.2 Computation of the Different Metrics

The following lemma details the computation of the conditional survival function.

Lemma 11. For any $i \in \{1, \dots, n\}$,

$$r_\theta^i(t, \delta t) = \exp\Big(- \int_t^{t+\delta t} \lambda_\theta^i(u, x_{[0,u\wedge t]}^{i,D}) du \Big),$$

where $r_\theta^i(t, \delta t) = \mathbb{P}\big(T^i > t + \delta t \,|\, T^i > t, x_{[0,t]}^{i,D}\big)$ is the survival function of individual i, as estimated by the model with parameters θ, at time $t + \delta t$ for $\delta t > 0$ conditional on survival up to time t, and on observation of the longitudinal features up to time t, and the notation $\lambda_\theta^i(u, x_{[0,u\wedge t]}^{i,D})$ means that the intensity at time u is computed by using the longitudinal features up to time $u \wedge t = \min(u, t)$.

Proof. Since Bayes rule gives

$$r_\theta^i(t, \delta t) = \mathbb{P}\Big(T^i > t + \delta t \,|\, T^i > t, x_{[0,t]}^{i,D}, \mathbf{W}^i\Big) = \frac{\mathbb{P}\Big(T^i > t + \delta t \,|\, x_{[0,t]}^{i,D}, \mathbf{W}^i\Big)}{\mathbb{P}\Big(T^i > t \,|\, x_{[0,t]}^{i,D}, \mathbf{W}^i\Big)},$$

we can compute this score by using the fact that

$$\mathbb{P}\Big(T^i > t \,|\, x_{[0,t]}^{i,D}, \mathbf{W}^i\Big) = \exp(-\Lambda_\theta^{i,D}(t)),$$

where we recall that $\Lambda_\theta^{i,D}(t)$ is the cumulative hazard function

$$\Lambda_\theta^{i,D}(t) := \int_0^t \lambda_\theta^{i,D}(s) Y^i(s) ds.$$

We refer the reader unfamiliar with survival analysis to Aalen et al. (2008, Chapter 1, p. 6) for a proof of this expression of the survival function. This then yields

$$\begin{aligned} r_\theta^i(t, \delta t) &= \frac{\exp(-\int_0^{t+\delta t} \lambda_\theta^i(u, x_{[0,u\wedge t]}^{i,D}) du)}{\exp(-\int_0^t \lambda_\theta^i(u, x_{[0,u\wedge t]}^{i,D}) du)} \\ &= \exp(-\int_t^{t+\delta t} \lambda_\theta^i(u, x_{[0,u\wedge t]}^{i,D}) du). \end{aligned}$$

Beside the two metrics described in the main paper, we report our results in term of two more metrics namely the weighted Brier Score and the area under the receiver operating characteristic curve (AUC). The details of these metrics are given below.

Weighted Brier Score. The weighted version of the Brier score, which we write $\text{WBS}(t, \delta t)$, is defined as

$$\sum_{i=1}^{n} \mathbb{1}_{T^i \leq t,\, \Delta^i=1} \frac{r_\theta^i(t, \delta t))^2}{\hat{G}(T^i)} + \mathbb{1}_{T^i \geq t} \frac{(1 - r_\theta^i(t, \delta t))^2}{\hat{G}(t)},$$

where $\hat{G}(\cdot)$ is the probability of censoring weight, estimated by the Kaplan-Meier estimator.

AUC. We define the area under the receiver operating characteristic curve $\text{AUC}(t, \delta t)$ as

$$\frac{\sum_{i=1}^{n} \sum_{j=1}^{n} \mathbb{1}_{r_\theta^i(t,\delta t) > r_\theta^j(t,\delta t)} \mathbb{1}_{T^i > t+\delta t,\, T^j \in [t, t+\delta t]} w_j}{(\sum_{i=1}^{n} \mathbb{1}_{T^i > t+\delta t})(\sum_{i=1}^{n} \mathbb{1}_{T^i \in [t, t+\delta t]} w_i)},$$

where w_i are inverse probability of censoring weights, estimated by the Kaplan-Meier estimator.

B.4 Details of Experiments and Datasets

The main characteristics of the datasets used in the paper are summarized in Table B.1 and we provide more detailed information of these datasets in subsections below. For the experiments, each dataset is randomly divided into a training set (80%) and test set (20%). Hyperparameter optimization is performed as follows. We split the training set, using 4/5 for training and 1/5 for validation. We then re-fit on the whole training set with the best hyperparameters and report the results on the test set for 10 runs. Note that the performance is evaluated at numerous points (t, δ_t), where t is set to the 5th, 10th, and 20th percentile of the distribution of event times.

Name	n	d	Static Features	Censoring	Avg. Observation Times	Source
Hitting time	500	5	✗	Terminal (3.2%)	177	Simulation
Tumor Growth	500	2	✗	Terminal (8.4%)	250	Simeoni et al. (2004)
Predictive Maintenance	200	17	✗	Online (50%)	167	Saxena et al. (2008)
Churn	1043	14	✗	Terminal (38.4%)	25	Private dataset

Table B.1 – Description of the 4 datasets we consider. The integer d is the dimension of the time series including the time channel. *Terminal* censoring means that the individuals are censored at the end of the overall observation period $[0, \tau]$ if they have not experienced any event. It is opposed to *online* censoring that can happen at any time in $[0, \tau]$. The reported percentage indicates the censoring level i.e. the share of the population that does not experience the event.

B.4.1 Hitting Time of a partially observed SDE

Time series. The paths $x_t = (x_t^{(1)}, \ldots, x_t^{(d-1)})$ are $(d-1)$-dimensional sample paths of a fractional Brownian motion with Hurst parameter $H = 0.6$, and $B^i(t)$ is a Brownian noise term. We set $d = 5$. The paths are sampled at 1000 times over the time interval $[0, 10]$. All simulations are done using the `stochastic` package [1]. The time series $\mathbf{X}^i$ are identical, up to observation time, to the ones used for simulations.

Event definition We consider the stochastic differential equation

$$dw_t = -\omega(w_t - \mu)dt + \sum_{i=1}^{d} dx_t^{(i)} + \sigma dB_t,$$

where w_t is trajectory of each individual with $(\sigma, \mu, \omega) \in \mathbb{R}^3$ are fixed parameters. In our experiment, the parameters are chosen to be $\sigma = 1$, $\mu = 0.1$ and $\omega = 0.1$. We then define the time-of-event as the time when trajectory cross the threshold $w_\star \in \mathbb{R}$ during the observation period $[t_0\ t_N]$, which is

$$T^\star = \min\{t_0 \leq t \leq t_N \,|\, w_t \geq w_\star\}.$$

In our experiments, we use the threshold value $w_\star = 2.5$. The target SDE is simulated using an Euler discretization. We train on $n = 500$ individuals.

Censorship We censor individuals whose trajectory does not cross the threshold during the observation period. This means that individuals are never censored during the observation period, but only at the end. The simulated censoring level is 3.2%.

Supplementary Figures. Figure B.3 provides an example of the full sample path of an individual and the distribution of the event times of the whole population. We add additional results on the test set in Figures B.4, B.5, B.6, B.7 and B.8.

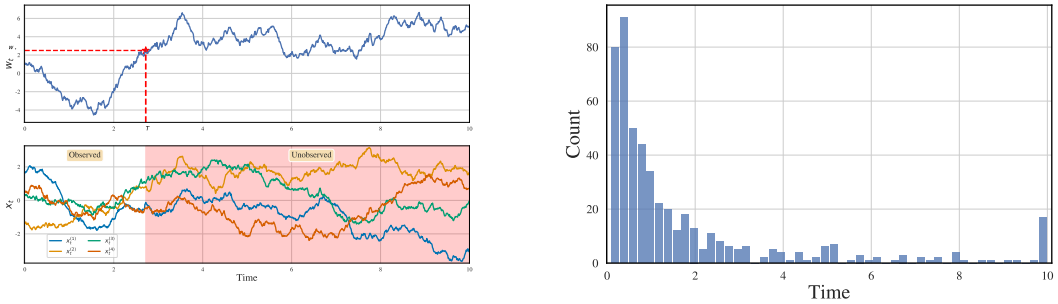

Figure B.3 – Full sample path of an individual (**left**) and distribution of the event times (**left**) for the partially observed SDE experiment. The surge in events at the terminal time indicates terminal censorship.

1. Available at https://github.com/crflynn/stochastic

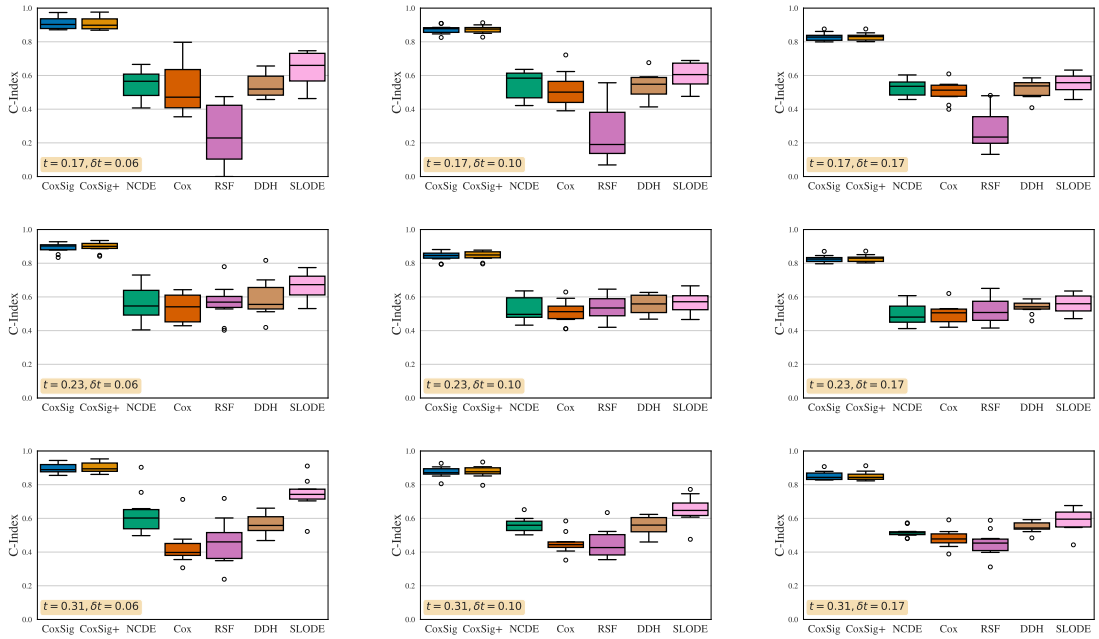

Figure B.4 – C-Index (*higher* is better) for **hitting time of a partially observed SDE** for numerous points $(t, \delta t)$.

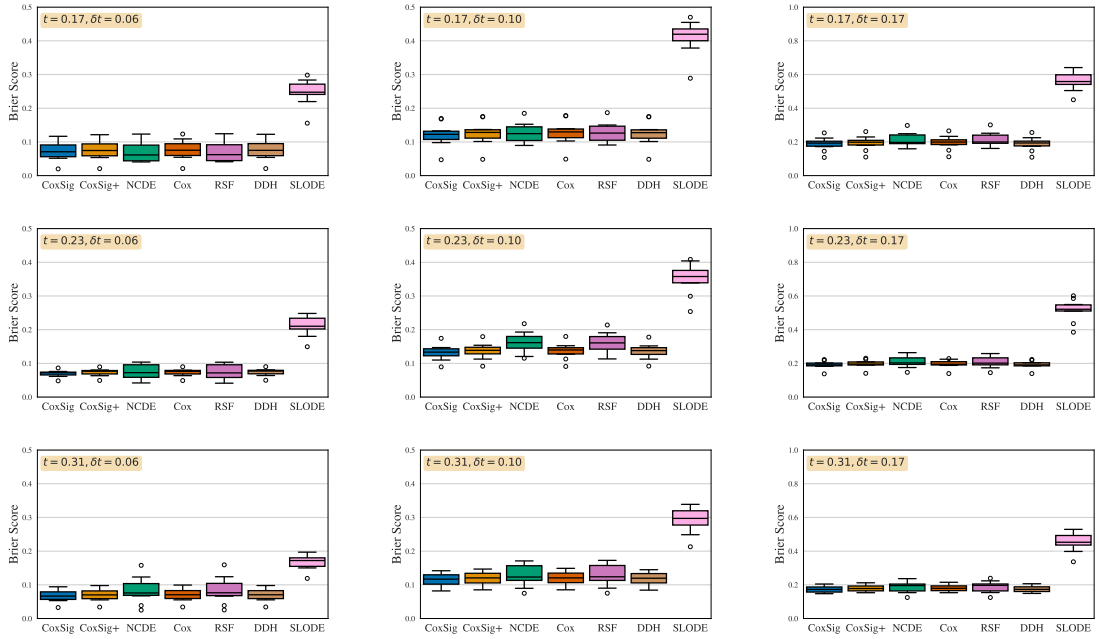

Figure B.5 – Brier score (*lower* is better) for **hitting time of a partially observed SDE** for numerous points $(t, \delta t)$.

B.4.2 Tumor Growth

Time series. Similarly to the partially observed SDE experiment described above, we set $d = 2$ which includes 1-dimensional sample path x_t of a fractional Brownian motion with

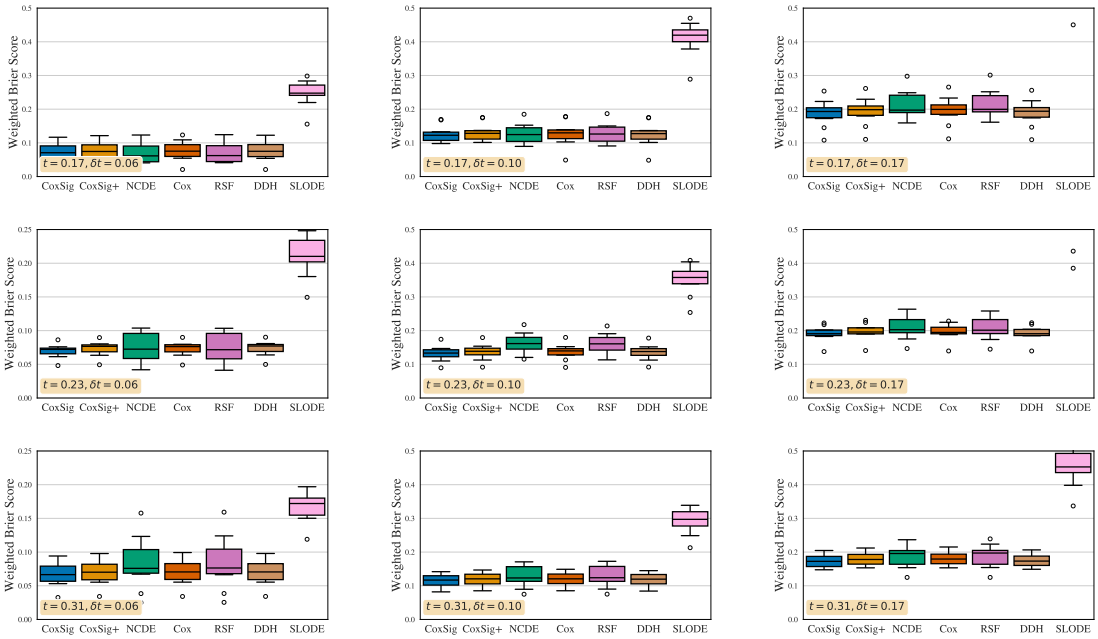

Figure B.6 – Weighted Brier score (*lower* is better) for **hitting time of a partially observed SDE** for numerous points $(t, \delta t)$.

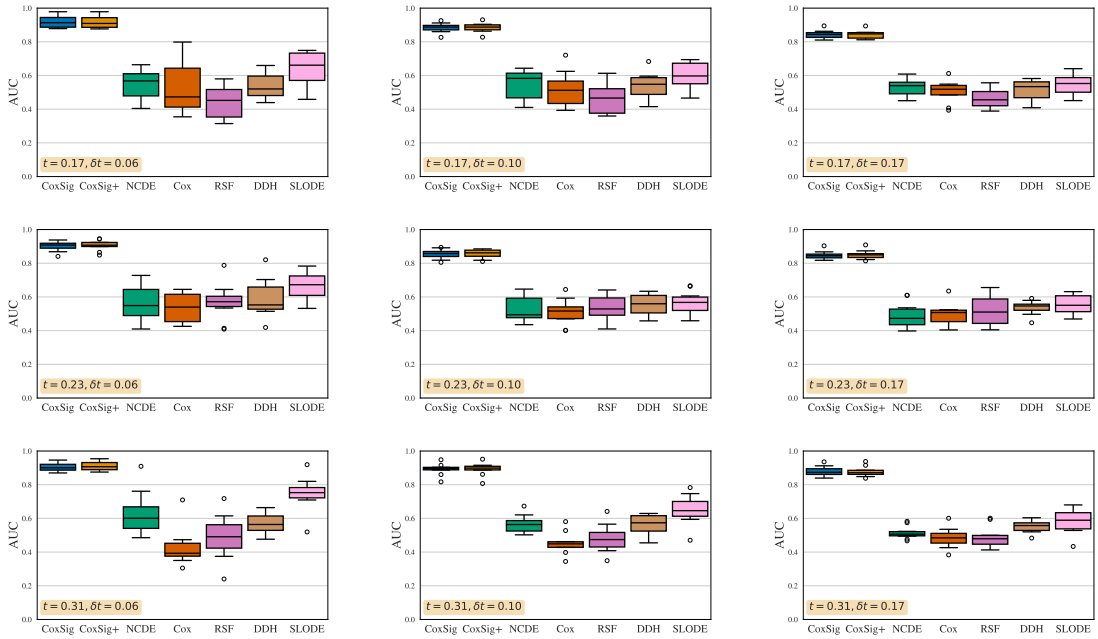

Figure B.7 – AUC (*higher* is better) for **hitting time of a partially observed SDE** for numerous points $(t, \delta t)$.

Hurst parameter $H = 0.6$. The paths are sampled at 1000 times over the time interval $[0, 10]$. All simulations are done using the `stochastic` package. The time series $\mathbf{X}^i$ are identical, up to observation time, to the ones used for simulations.

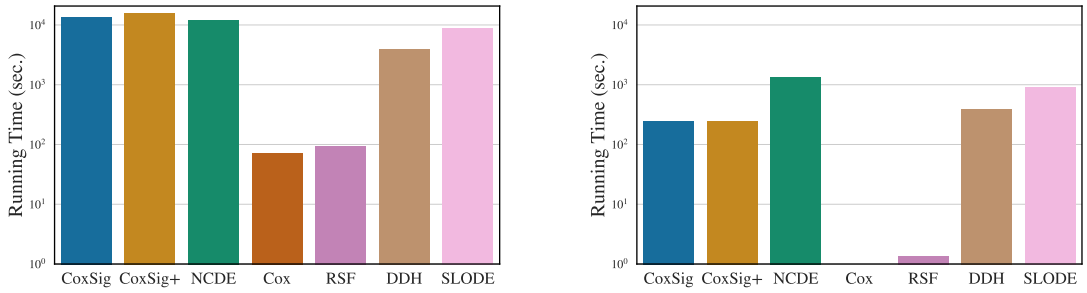

Figure B.8 – Running times on the partially observed SDE experiment (log-scale) averaged over 10 runs including cross-validation of the hyperparameters on CoxSig, CoxSig+, Cox and RSF (**left**) and over 1 run without cross-validation of the hyperparameters on CoxSig, CoxSig+, Cox and RSF (**right**).

Event definition. Following Simeoni et al. (2004), we consider the differential equations

$$\begin{aligned}
\frac{du_t^{(1)}}{dt} &= \frac{\lambda_0 u_t^{(1)}}{\left[1 + (\frac{\lambda_0}{\lambda_1} w_t)^{\Psi}\right]^{1/\Psi}} - \kappa_2 x_t u_t^{(1)} \\
\frac{du_t^{(2)}}{dt} &= \kappa_2 x_t u_t^{(1)} - \kappa_1 u_t^{(2)} \\
\frac{du_t^{(3)}}{dt} &= \kappa_1 (u_t^{(2)} - u_t^{(3)}) \\
\frac{du_t^{(4)}}{dt} &= \kappa_1 (u_t^{(3)} - u_t^{(4)}) \\
w_t &= u_t^{(1)} + u_t^{(2)} + u_t^{(3)} + u_t^{(4)},
\end{aligned}$$

where w_t is trajectory of each individual with initial status of $(u_0^{(1)}, u_0^{(2)}, u_0^{(3)}, u_0^{(4)}) = (0.8, 0, 0, 0)$ and $(\lambda_0, \lambda_1, \kappa_1, \kappa_2, \Psi) \in \mathbb{R}^5$ are fixed parameters. In our experiment, the parameters are chosen to be $\lambda_0 = 0.9$, $\lambda_1 = 0.7$, $\kappa_1 = 10$, $\kappa_2 = 0.15$ and $\Psi = 20$. We then define the time-of-event as the time when trajectory cross the threshold $w_\star \in \mathbb{R}$ during the observation period $[t_0\ t_N]$, which is

$$T^\star = \min\{t_0 \leq t \leq t_N \,|\, w_t \geq w_\star\}.$$

In our experiments, we use the threshold value $w_\star = 1.7$. The target differential equations are simulated using an Euler discretization. We train on $n = 500$ individuals.

Censorship. Similarly to the partially observed SDE experiment, we consider terminal censorship: individuals that do not experience the event within the observation period are censored. The censoring level is 8.4%.

Supplementary Figures. Figure B.9 provides an example of the full sample path of an individual and the distribution of the event times of the whole population. We add additional

results on the test set in Figures B.10, B.11, B.12 and B.13.

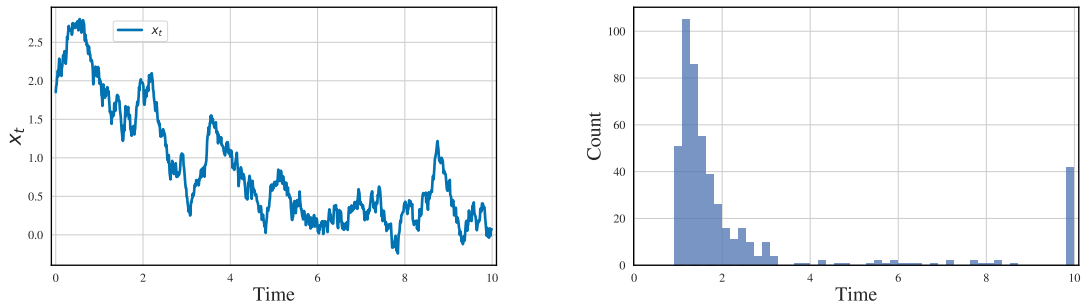

Figure B.9 – Full sample path of an individual (**left**) and distribution of the event times (**left**) for the tumor growth experiment. The surge in events at the terminal time indicates terminal censorship.

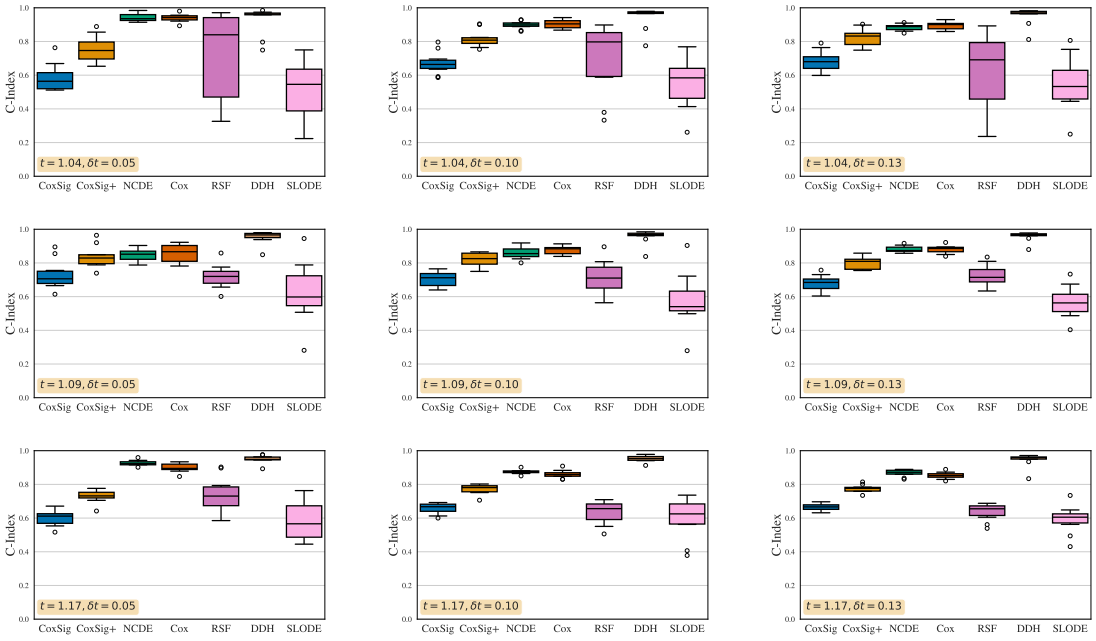

Figure B.10 – C-Index (*higher* is better) for **Tumor Growth** for numerous points $(t, \delta t)$.

B.4.3 Predictive Maintenance

Time series. This dataset describes the degradation of 200 aircraft gas turbine engines, where 22 measurements of sensors and 3 operational settings are recorded each operational cycle until its failure. After removing low-variance features, 16 longitudinal features are selected for training models. The average time length of these features is about 25 cycles. Note that we apply standardization for selected features before training.

Event definition. The times of event are given as-is in the dataset. We refer to Saxena et al. (2008) for a precise description of the data generation.

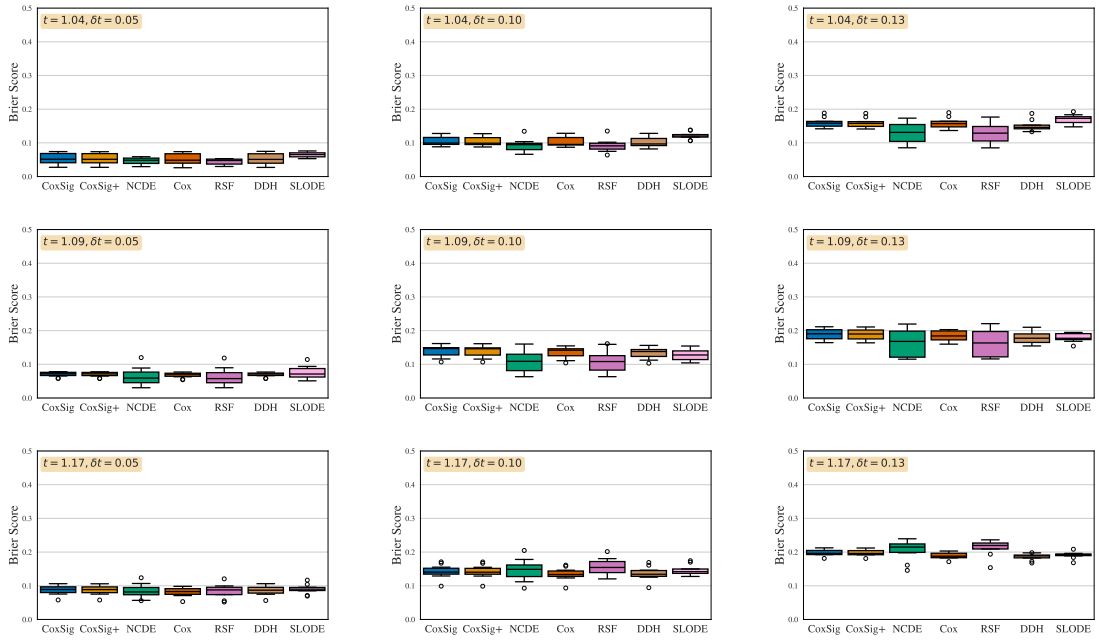

Figure B.11 – Brier score (*lower* is better) for **Tumor Growth** for numerous points $(t, \delta t)$.

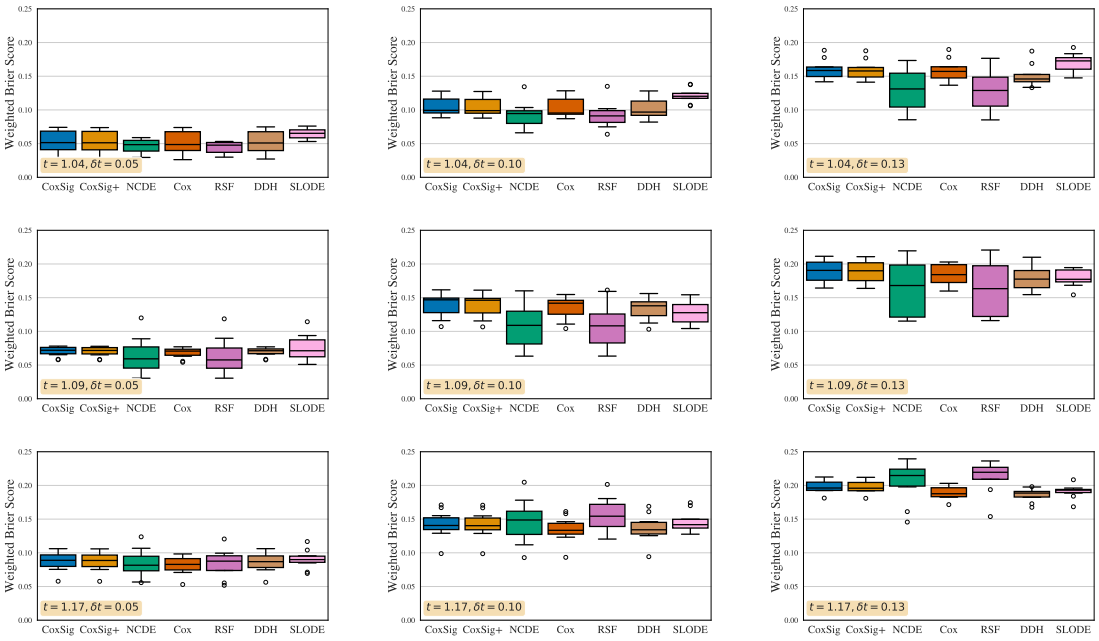

Figure B.12 – Weighted Brier score (*lower* is better) for **Tumor Growth** for numerous points $(t, \delta t)$.

Censorship. Censorship is given as-in in the dataset. The censoring level of this dataset is 50%, which is a high censorship rate in survival analysis. We refer again to Saxena et al. (2008) for more details.

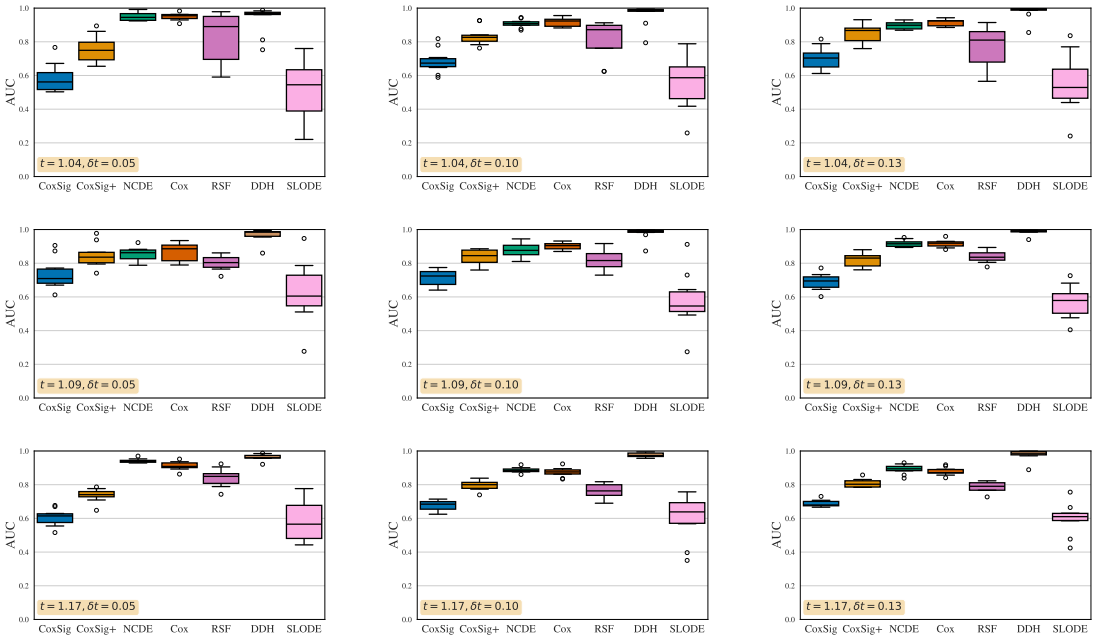

Figure B.13 – AUC (*higher* is better) for **Tumor Growth** for numerous points $(t, \delta t)$.

Supplementary Figures. Figure B.14 provides an example of several randomly picked sample paths of an individual and the distribution of the event times of the whole population. We add additional results in Figures B.15, B.16, B.17 and B.18.

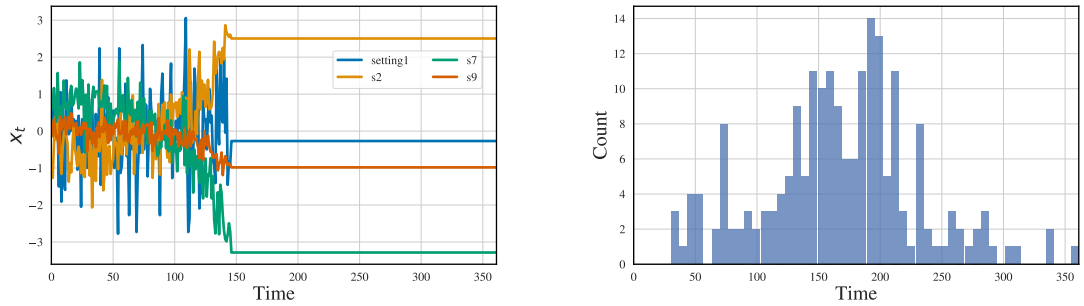

Figure B.14 – Partial sample path of an individual (**left**) and distribution of the event times (**left**) for the predictive maintenance experiment. On the left, the time series is filled with the last observed value from the time of the event on.

B.4.4 Churn Prediction

For this dataset, the amount of details that we can release is limited both because of the sensitive nature of the data and of the anonymity requirements of the reviewing process.

Time series. All longitudinal features have been computed on a temporal window of one week, the raw data corresponding to all product orders placed on the platform from 06-12-

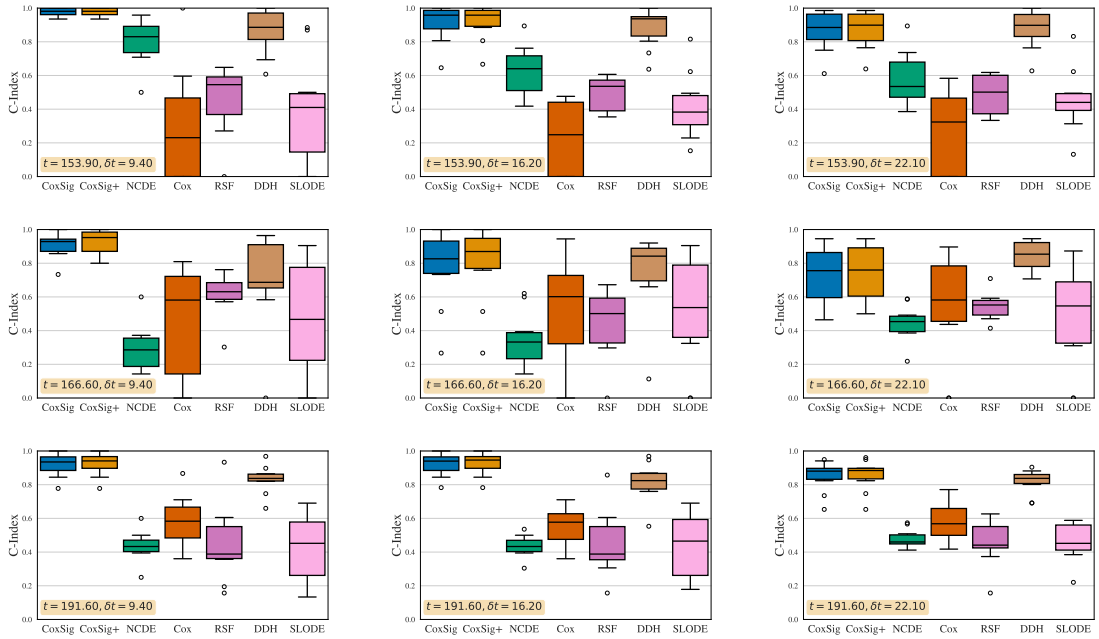

Figure B.15 – C-Index (*higher* is better) for **predictive maintenance** for numerous points $(t, \delta t)$.

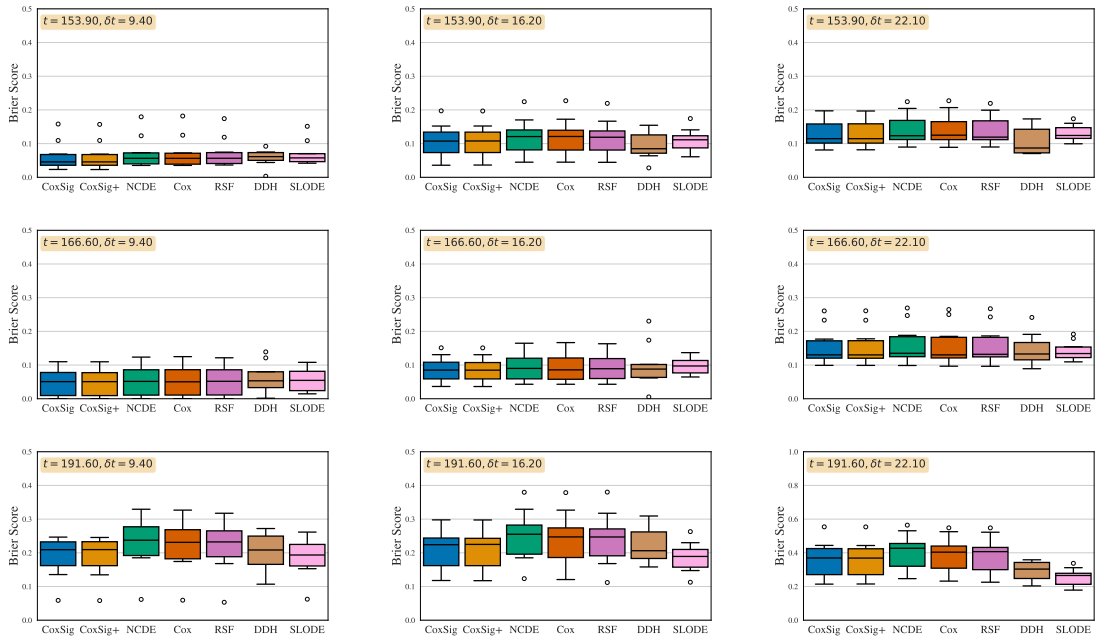

Figure B.16 – Brier Score (*lower* is better) for **predictive maintenance** for numerous points $(t, \delta t)$.

2021 to 12-11-2023. For clients who have no order during the week, we fill zero value for all longitudinal measurements this week. After removing features with more than 90 % of missingness, 14 longitudinal features of 1043 clients are selected for the training step. Note that we apply standardization for selected features before training.

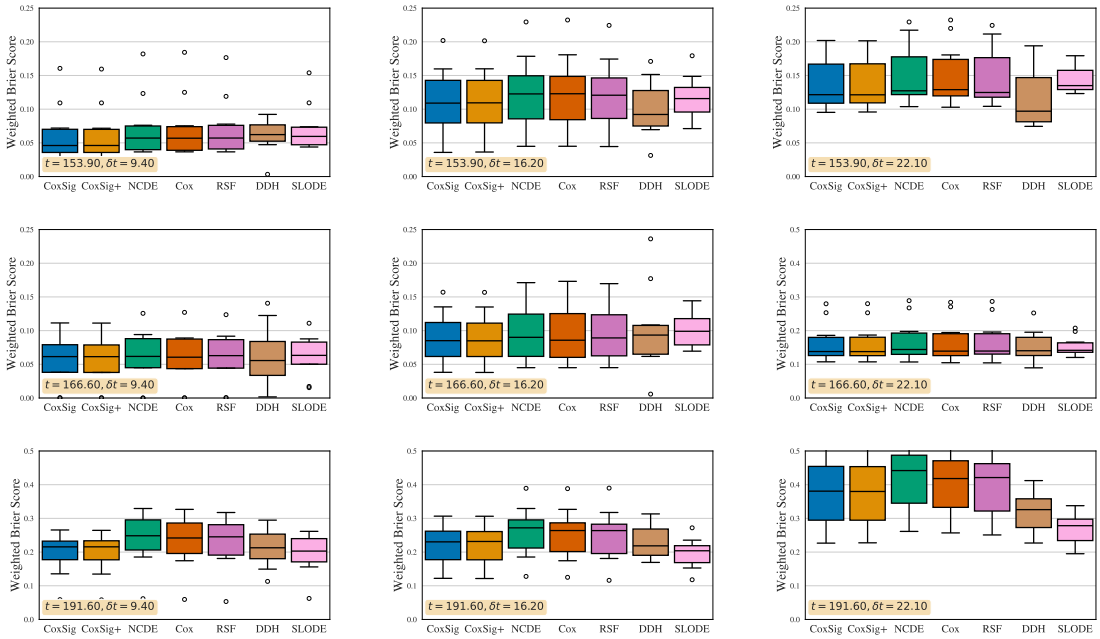

Figure B.17 – Weighted Brier Score (*lower* is better) for **predictive maintenance** for numerous points $(t, \delta t)$.

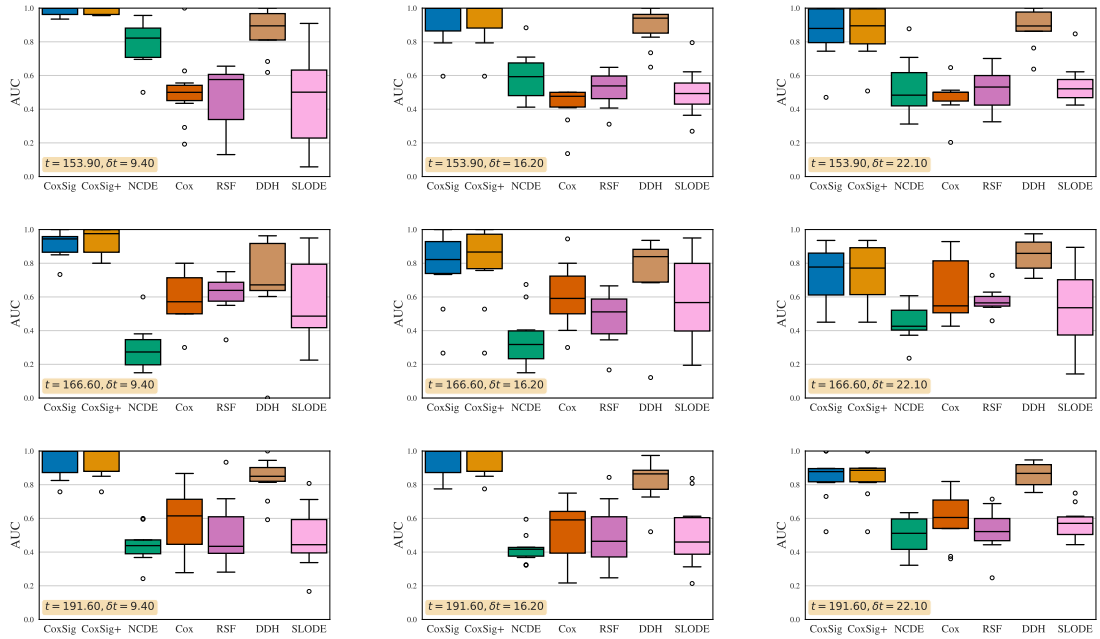

Figure B.18 – AUC (*higher* is better) for **predictive maintenance** for numerous points $(t, \delta t)$.

Event definition. We consider that a customer has churned if she has no passed any order in the last 4 weeks. If the customer starts ordering again after a churn, we register her as a new customer.

Censorship. Censorship is terminal based on the data collection period (give dates here). Hence any customer that has not churned by 12-11-2023 is censored. In this dataset, 38.4% of the clients are terminally censored.

Supplementary Figures. Figure B.19 provides an example of four sample paths of four randomly chosen individuals. We add additional results in Figures B.20, B.21, B.22 and B.23.

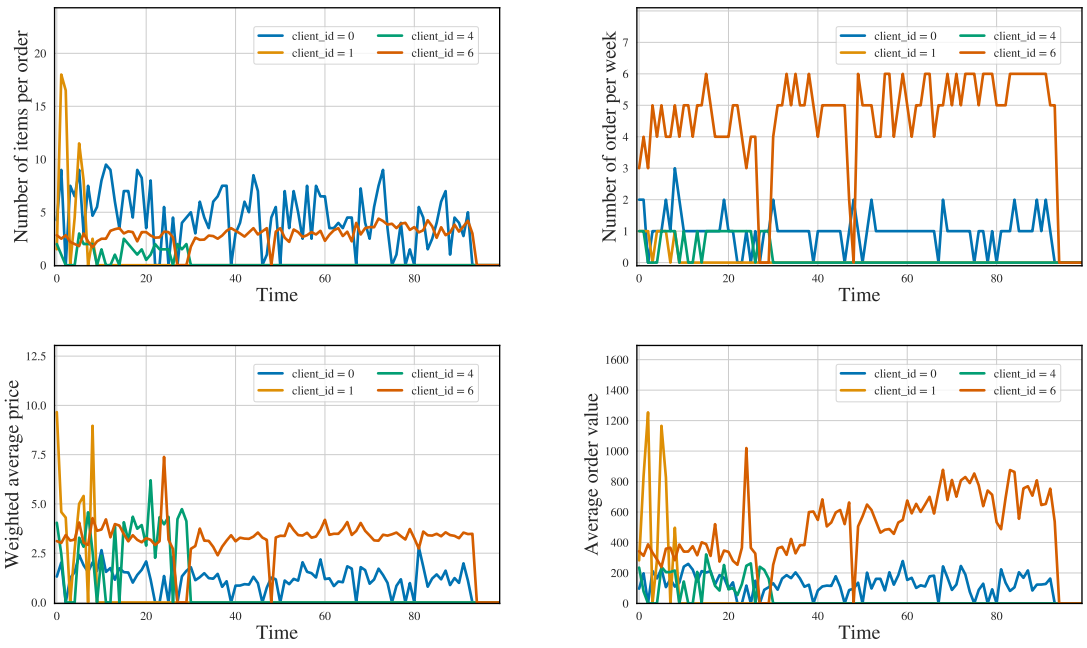

Figure B.19 – Values of 4 different time-dependent features for 4 randomly chosen individuals from the **churn prediction** dataset. Individual time-to-event and distribution of the event times cannot be displayed to protect consumer and business privacy. A precise description of the different time-dependent features will be provided upon publication.

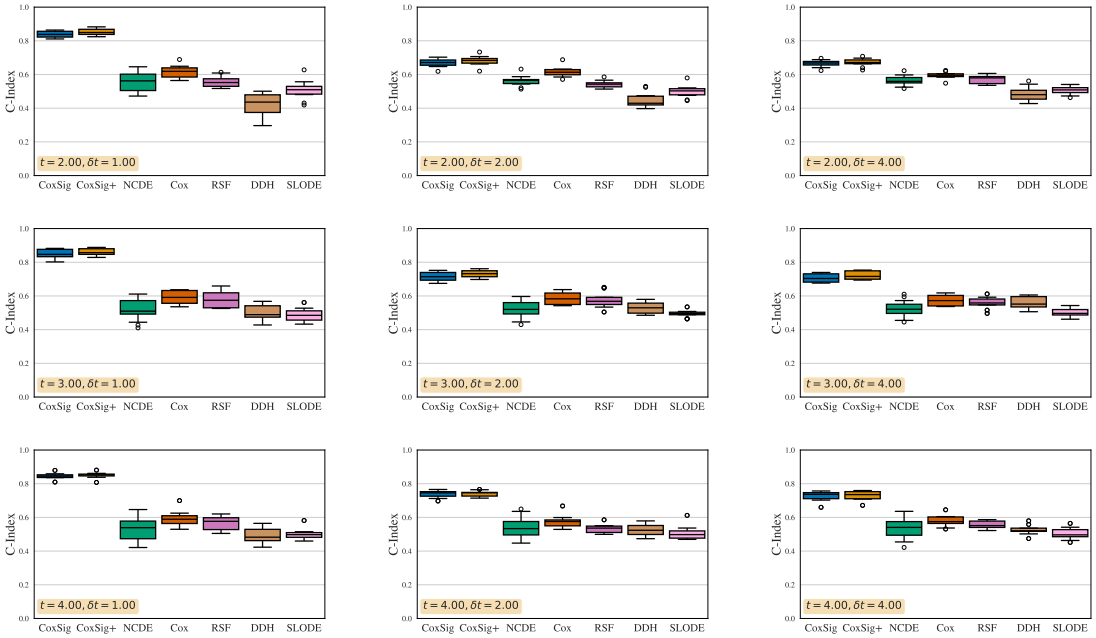

Figure B.20 – C-Index (*higher* is better) for **churn prediction** for numerous points $(t, \delta t)$.

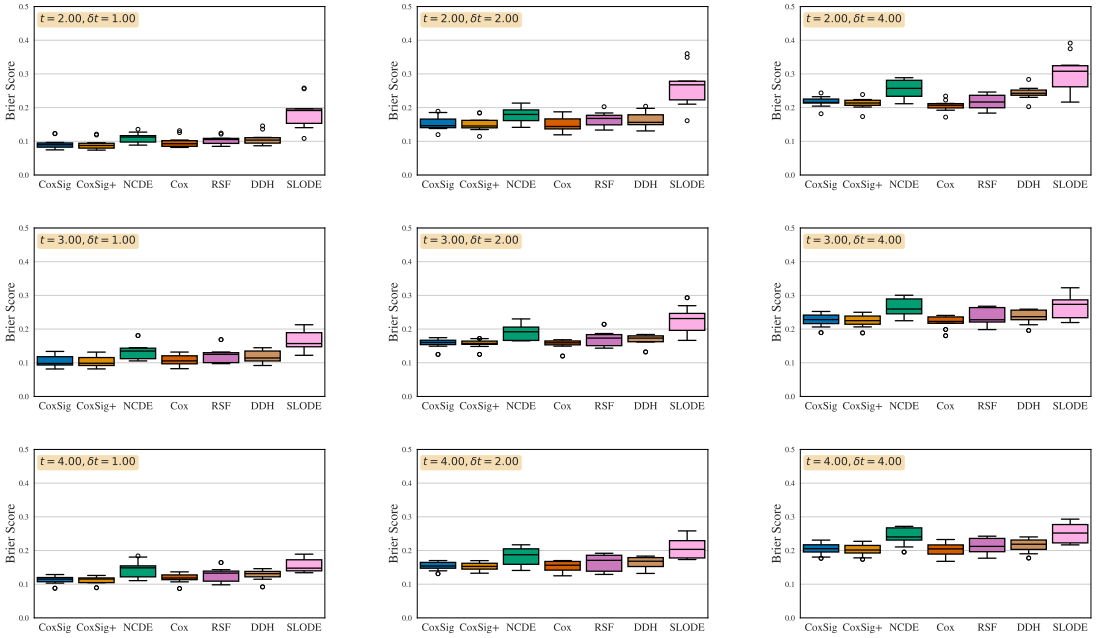

Figure B.21 – Brier score (*lower* is better) for **churn prediction** for numerous points $(t, \delta t)$.

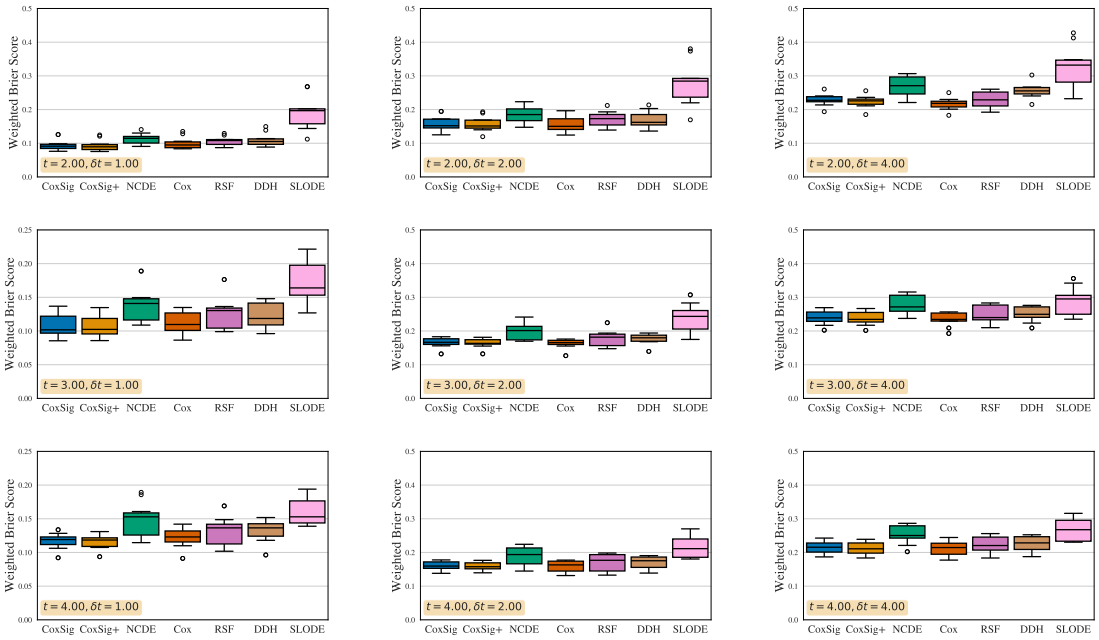

Figure B.22 – Weighted Brier score (*lower* is better) for **churn prediction** for numerous points $(t, \delta t)$.

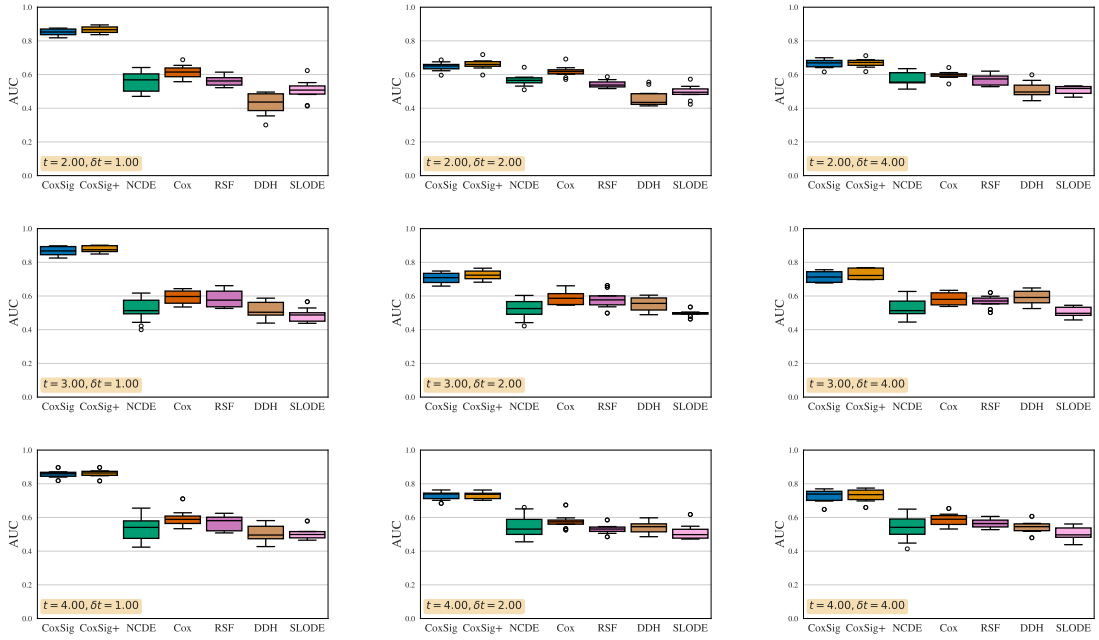

Figure B.23 – AUC (*higher* is better) for **churn prediction** for numerous points $(t, \delta t)$.

Bibliography

Aalen, Odd (1978). « Nonparametric inference for a family of counting processes ». In: *The Annals of Statistics*, pp. 701–726.

Aalen, Odd, Ornulf Borgan, and Hakon Gjessing (2008). *Survival and event history analysis: a process point of view*. Springer Science & Business Media.

Aalen, Odd O, Per Kragh Andersen, Ørnulf Borgan, Richard D Gill, and Niels Keiding (2010). « History of applications of martingales in survival analysis ». In: *arXiv preprint arXiv:1003.0188*.

Ahn, Jaehyun, Junsik Hwang, Doyoung Kim, Hyukgeun Choi, and Shinjin Kang (2020). « A survey on churn analysis in various business domains ». In: *IEEE Access* 8, pp. 220816–220839.

Alboukaey, Nadia, Ammar Joukhadar, and Nada Ghneim (2020). « Dynamic behavior based churn prediction in mobile telecom ». In: *Expert Systems with Applications* 162, p. 113779.

Aldrich, John H and Forrest D Nelson (1984). *Linear probability, logit, and probit models*. 45. Sage.

Andersen, Per K, Ornulf Borgan, Richard D Gill, and Niels Keiding (2012). *Statistical models based on counting processes*. Springer Science & Business Media.

Andersen, Per Kragh and Richard D Gill (1982). « Cox's regression model for counting processes: a large sample study ». In: *The annals of statistics*, pp. 1100–1120.

Anderson, James R, Kevin C Cain, Richard D Gelber, et al. (1983). « Analysis of survival by tumor response ». In: *J Clin Oncol* 1.11, pp. 710–719.

Andrew, Galen and Jianfeng Gao (2007). « Scalable training of L1-regularized log-linear models ». In: *International Conference on Machine Learning*. ACM, pp. 33–40.

Andrinopoulou, Eleni-Rosalina, Kazem Nasserinejad, Rhonda Szczesniak, and Dimitris Rizopoulos (2020). « Integrating latent classes in the Bayesian shared parameter joint model of longitudinal and survival outcomes ». In: *Statistical methods in medical research* 29.11, pp. 3294–3307.

Andrinopoulou, Eleni-Rosalina and Dimitris Rizopoulos (2016). « Bayesian shrinkage approach for a joint model of longitudinal and survival outcomes assuming different association structures ». In: *Statistics in medicine* 35.26, pp. 4813–4823.

Athanassopoulos, Antreas D (2000). « Customer satisfaction cues to support market segmentation and explain switching behavior ». In: *Journal of business research* 47.3, pp. 191–207.

Austin, Peter C (2013). « Correction:‘Generating survival times to simulate Cox proportional hazards models with time-varying covariates’ ». In: *Statistics in Medicine* 32.6, pp. 1078–1078.

Avati, Anand, Tony Duan, Sharon Zhou, Kenneth Jung, Nigam H Shah, and Andrew Y Ng (2020). « Countdown regression: sharp and calibrated survival predictions ». In: *Uncertainty in Artificial Intelligence*. PMLR, pp. 145–155.

Bach, Francis (2021). « Learning theory from first principles ». In: *Draft of a book, version of Sept* 6, p. 2021.

— (2010). « Self-concordant analysis for logistic regression ». In: *Electronic Journal of Statistics* 4, pp. 384–414.

Bach, Francis, Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, et al. (2012). « Optimization with sparsity-inducing penalties ». In: *Foundations and Trends® in Machine Learning* 4.1, pp. 1–106.

Bach, Francis R (2008). « Bolasso: model consistent lasso estimation through the bootstrap ». In: *Proceedings of the 25th international conference on Machine learning*, pp. 33–40.

Bacry, Emmanuel, Iacopo Mastromatteo, and Jean-François Muzy (2015). « Hawkes processes in finance ». In: *Market Microstructure and Liquidity* 1.01, p. 1550005.

Bartolucci, Francesco and Alessio Farcomeni (2015). « A discrete time event-history approach to informative drop-out in mixed latent Markov models with covariates ». In: *Biometrics* 71.1, pp. 80–89.

— (2019). « A shared-parameter continuous-time hidden Markov and survival model for longitudinal data with informative dropout ». In: *Statistics in medicine* 38.6, pp. 1056–1073.

Beck, Amir and Marc Teboulle (2009). « A fast iterative shrinkage-thresholding algorithm for linear inverse problems ». In: *SIAM journal on imaging sciences* 2.1, pp. 183–202.

Bender, Ralf, Thomas Augustin, and Maria Blettner (2005). « Generating survival times to simulate Cox proportional hazards models ». In: *Statistics in medicine* 24.11, pp. 1713–1723.

Bertens, Paul, Anna Guitart, and África Periáñez (2017). « Games and big data: A scalable multi-dimensional churn prediction model ». In: *2017 IEEE conference on computational intelligence and games (CIG)*. IEEE, pp. 33–36.

Biganzoli, Elia, Patrizia Boracchi, Luigi Mariani, and Ettore Marubini (1998). « Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach ». In: *Statistics in medicine* 17.10, pp. 1169–1186.

Bland, J Martin and Douglas G Altman (2004). « The logrank test ». In: *Bmj* 328.7447, p. 1073.

Bleistein, Linus, Adeline Fermanian, Anne-Sophie Jannot, and Agathe Guilloux (2023). « Learning the Dynamics of Sparsely Observed Interacting Systems ». In: *40th International Conference on Machine Learning*.

Bleistein, Linus and Agathe Guilloux (2024). « On the Generalization Capacities of Neural Controlled Differential Equations ». In: *International Conference on Learning Representations*.

Bleistein, Linus, Van-Tuan Nguyen, Adeline Fermanian, and Agathe Guilloux (2024). « Dynamical Survival Analysis with Controlled Latent States ». In: *41th International Conference on Machine Learning and arXiv preprint arXiv:2401.17077*.

Böhning, Dankmar (1992). « Multinomial logistic regression algorithm ». In: *Annals of the institute of Statistical Mathematics* 44.1, pp. 197–200.

Bottou, Léon (2012). « Stochastic gradient descent tricks ». In: *Neural Networks: Tricks of the Trade: Second Edition*. Springer, pp. 421–436.

Boyd, Stephen and Lieven Vandenberghe (2004). *Convex optimization*. New York: Cambridge university press.

Breiman, Leo (2001). « Random forests ». In: *Machine learning* 45, pp. 5–32.

Breslow, Norman E (1972). « Contribution to discussion of paper by DR Cox ». In: *Journal of the Royal Statistical Society, Series B* 34, pp. 216–217.

Brier, Glenn W (1950). « Verification of forecasts expressed in terms of probability ». In: *Monthly weather review* 78.1, pp. 1–3.

Buckinx, Wouter and Dirk Van den Poel (2005). « Customer base analysis: partial defection of behaviourally loyal clients in a non-contractual FMCG retail setting ». In: *European journal of operational research* 164.1, pp. 252–268.

Buis, Maarten L (2006). « An introduction to survival analysis ». In: *Department of Social Research Methodology Vrije Universiteit Amsterdam [Online]*.

Burez, Jonathan and Dirk Van den Poel (2007). « CRM at a pay-TV company: Using analytical models to reduce customer attrition by targeted marketing for subscription services ». In: *Expert Systems with Applications* 32.2, pp. 277–288.

Bussy, Simon, Agathe Guilloux, Stéphane Gaïffas, and Anne-Sophie Jannot (2019a). « C-mix: A high-dimensional mixture model for censored durations, with applications to genetic data ». In: *Statistical methods in medical research* 28.5, pp. 1523–1539.

Bussy, Simon, Raphaël Veil, Vincent Looten, Anita Burgun, Stéphane Gaïffas, Agathe Guilloux, Brigitte Ranque, and Anne-Sophie Jannot (2019b). « Comparison of methods for early-readmission prediction in a high-dimensional heterogeneous covariates and time-to-event outcome framework ». In: *BMC medical research methodology* 19, pp. 1–9.

Candes, Emmanuel and Benjamin Recht (2012). « Exact matrix completion via convex optimization ». In: *Communications of the ACM* 55.6, pp. 111–119.

Chen, Kuo-Tsai (1958). « Integration of paths—a faithful representation of paths by noncommutative formal power series ». In: *Transactions of the American Mathematical Society* 89, pp. 395–407.

Chen, Ricky T. Q., Brandon Amos, and Maximilian Nickel (2021). « Neural Spatio-Temporal Point Processes ». In: *International Conference on Learning Representations*.

Chen, Ricky T. Q., Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud (2018). « Neural ordinary differential equations ». In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc., pp. 6572–6583.

Cheridito, Patrick and Zhikai Xu (2015). « Pricing and hedging CoCos ». In: *Available at SSRN 2201364*.

Chi, Yueh-Yun and Joseph G Ibrahim (2006). « Joint models for multivariate longitudinal and multivariate survival data ». In: *Biometrics* 62.2, pp. 432–445.

Christ, Maximilian, Nils Braun, Julius Neuffer, and Andreas W Kempa-Liehr (2018). « Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package) ». In: *Neurocomputing* 307, pp. 72–77.

Chzhen, Evgenii, Mohamed Hebiri, and Joseph Salmon (2019). « On lasso refitting strategies ». In.

Cirone, Nicola Muca, Maud Lemercier, and Cristopher Salvi (2023). « Neural signature kernels as infinite-width-depth-limits of controlled resnets ». In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett. PMLR, pp. 25358–25425.

Cirone, Nicola Muca, Antonio Orvieto, Benjamin Walker, Cristopher Salvi, and Terry Lyons (2024). « Theoretical Foundations of Deep Selective State-Space Models ». In: *arXiv preprint arXiv:2402.19047*.

Corcuera, José Manuel and Arturo Valdivia (2016). « Pricing CoCos with a Market Trigger ». In: *Stochastics of Environmental and Financial Economics*. Ed. by Fred Espen Benth and Giulia Di Nunno. Springer International Publishing, pp. 179–209.

Coussement, Kristof and Dirk Van den Poel (2008). « Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques ». In: *Expert systems with applications* 34.1, pp. 313–327.

Cox, D. R. (1972a). « Regression Models and Life-Tables ». In: *Journal of the Royal Statistical Society. Series B (Methodological)* 34.2, pp. 187–220.

Cox, David R (1972b). « Regression models and life-tables ». In: *Journal of the Royal Statistical Society: Series B (Methodological)* 34.2, pp. 187–202.

Crowther, Michael J, Keith R Abrams, and Paul C Lambert (2013). « Joint modeling of longitudinal and survival data ». In: *The Stata Journal* 13.1, pp. 165–184.

De Boor, C (1978). « A practical guide to splines ». In: *Springer-Verlag google schola* 2, pp. 4135–4195.

De Brouwer, Edward, Javier Gonzalez, and Stephanie Hyland (2022). « Predicting the impact of treatments over time with uncertainty aware neural differential equations. » In: *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. Ed. by Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera. Vol. 151. PMLR, pp. 4705–4722.

De Brouwer, Edward, Jaak Simm, Adam Arany, and Yves Moreau (2019). « GRU-ODE-Bayes: Continuous modeling of sporadically-observed time series ». In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., pp. 7379–7390.

Dempster, AP, NM Laird, and DB Rubin (1977). « Maximum Likelihood from Incomplete Data via the EM Algorithm ». In: *Journal of the Royal Statistical Society. Series B (Methodological)* 39.1, pp. 1–38.

Devaux, Anthony, Robin Genuer, Karine Peres, and Cécile Proust-Lima (2022). « Individual dynamic prediction of clinical endpoint from large dimensional longitudinal biomarker history: a landmark approach ». In: *BMC Medical Research Methodology* 22.1, pp. 1–14.

Dumanis, Sonya B, Jaqueline A French, Christophe Bernard, Gregory A Worrell, and Brandy E Fureman (2017). « Seizure forecasting from idea to reality. Outcomes of the my seizure gauge epilepsy innovation institute workshop ». In: *Eneuro* 4.6.

Fabian Pedregosa Geoffrey Negiar, Gideon Dresdner (2020). « copt: composite optimization in Python ». In: URL: *http://openopt.github.io/copt/*.

Fabio, Lizandra C, Gilberto A Paula, and Mário de Castro (2012). « A Poisson mixed model with nonnormal random effect distribution ». In: *Computational Statistics & Data Analysis* 56.6, pp. 1499–1510.

Faraggi, David and Richard Simon (1995). « A neural network model for survival data ». In: *Statistics in medicine* 14.1, pp. 73–82.

Fermanian, A., P. Marion, J.-P. Vert, and G. Biau (2021). « Framing RNN as a kernel method: A neural ODE approach ». In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., pp. 3121–3134.

Fermanian, Adeline (2021). « Embedding and learning with signatures ». In: *Computational Statistics & Data Analysis* 157, p. 107148.

— (2022). « Functional linear regression with truncated signatures ». In: *Journal of Multivariate Analysis* 192, p. 105031.

Fitzmaurice, Garrett M, Nan M Laird, and James H Ware (2012). *Applied longitudinal analysis*. New Jersey: John Wiley & Sons.

Fotso, Stephane (2018). « Deep neural networks for survival analysis based on a multi-task framework ». In: *arXiv preprint arXiv:1801.05512*.

Friz, Peter K and Nicolas B Victoir (2010). *Multidimensional stochastic processes as rough paths: theory and applications*. Vol. 120. Cambridge University Press.

Gareth, James, Witten Daniela, Hastie Trevor, and Tibshirani Robert (2013). *An introduction to statistical learning: with applications in R*. Spinger.

Geller, Jay (2023). « Food and Drug Administration Published Final Guidance on Clinical Decision Support Software ». In: *Journal of Clinical Engineering* 48.1, pp. 3–7.

Gensheimer, Michael F and Balasubramanian Narasimhan (2019). « A scalable discrete-time survival model for neural networks ». In: *PeerJ* 7, e6257.

Gerds, Thomas A and Martin Schumacher (2006). « Consistent estimation of the expected Brier score in general survival models with right-censored event times ». In: *Biometrical Journal* 48.6, pp. 1029–1040.

Gompertz, Benjamin (1825). « XXIV. On the nature of the function expressive of the law of human mortality, and on a new mode of determining the value of life contingencies.

In a letter to Francis Baily, Esq. FRS &c ». In: *Philosophical transactions of the Royal Society of London* 115, pp. 513–583.

Gönen, Mithat and Glenn Heller (2005). « Concordance probability and discriminatory power in proportional hazards regression ». In: *Biometrika* 92.4, pp. 965–970.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep learning*. MIT press.

Groha, Stefan, Sebastian M Schmon, and Alexander Gusev (2020). « A general framework for survival analysis and multi-state modelling ». In: *arXiv preprint arXiv:2006.04893*.

Gu, Albert, Karan Goel, and Christopher Ré (2022). « Efficiently modeling long sequences with structured state spaces ». In: *International Conference on Learning Representations*.

Gui, Chun (2017). « Analysis of imbalanced data set problem: The case of churn prediction for telecommunication. » In: *Artif. Intell. Res.* 6.2, p. 93.

Gupta, Garima, Vishal Sunder, Ranjitha Prasad, and Gautam Shroff (2019). « CRESA: a deep learning approach to competing risks, recurrent event survival analysis ». In: *Advances in Knowledge Discovery and Data Mining: 23rd Pacific-Asia Conference, PAKDD 2019, Macau, China, April 14-17, 2019, Proceedings, Part II 23*. Springer, pp. 108–122.

Harrell, Frank E, Kerry L Lee, and Daniel B Mark (1996). « Tutorial in biostatistics multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors ». In: *Statistics in medicine* 15, pp. 361–387.

Harrell Jr, Frank E, Kerry L Lee, and Daniel B Mark (1996). « Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors ». In: *Statistics in medicine* 15.4, pp. 361–387.

Hastie, Trevor, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman (2009). *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer.

Hatfield, Laura A, Mark E Boye, and Bradley P Carlin (2011). « Joint modeling of multiple longitudinal patient-reported outcomes and survival ». In: *Journal of Biopharmaceutical Statistics* 21.5, pp. 971–991.

Heagerty, Patrick J, Thomas Lumley, and Margaret S Pepe (2000). « Time-dependent ROC curves for censored survival data and a diagnostic marker ». In: *Biometrics* 56.2, pp. 337–344.

Heagerty, Patrick J and Yingye Zheng (2005). « Survival model predictive accuracy and ROC curves ». In: *Biometrics* 61.1, pp. 92–105.

Hickey, Graeme L, Pete Philipson, Andrea Jorgensen, and Ruwanthi Kolamunnage-Dona (2018). « joineRML: a joint model and software package for time-to-event and multivariate longitudinal outcomes ». In: *BMC medical research methodology* 18, pp. 1–14.

— (2016). « Joint modelling of time-to-event and multivariate longitudinal outcomes: recent developments and issues ». In: *BMC medical research methodology* 16, pp. 1–15.

Horvath, Blanka, Maud Lemercier, Chong Liu, Terry Lyons, and Cristopher Salvi (2023). « Optimal Stopping via Distribution Regression: a Higher Rank Signature Approach ». In: *arXiv preprint arXiv:2304.01479*.

Ibrahim, Joseph G, Haitao Chu, and Liddy M Chen (2010). « Basic concepts and methods for joint models of longitudinal and survival data ». In: *Journal of Clinical Oncology* 28.16, p. 2796.

Ishwaran, Hemant, Udaya B Kogalur, Eugene H Blackstone, and Michael S Lauer (2008). « Random survival forests ». In.

Jaffa, Miran A, Mulugeta Gebregziabher, and Ayad A Jaffa (2014). « A Joint Modeling Approach for Right Censored High Dimensional Multivariate Longitudinal Data ». In: *Journal of biometrics & biostatistics* 5.4.

Jia, Junteng and Austin R Benson (2019). « Neural Jump Stochastic Differential Equations ». In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32.

Kalbfleisch, John D and Ross L Prentice (2002). « Competing risks and multistate models ». In: *The statistical analysis of failure time data* 247, p. 277.

Kang, Kai and Xinyuan Song (2022). « Consistent estimation of a joint model for multivariate longitudinal and survival data with latent variables ». In: *Journal of Multivariate Analysis* 187, p. 104827.

Kaplan, Edward L and Paul Meier (1958). « Nonparametric estimation from incomplete observations ». In: *Journal of the American statistical association* 53.282, pp. 457–481.

Katzman, Jared L, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger (2018). « DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network ». In: *BMC medical research methodology* 18, pp. 1–12.

Khodadadi, Ali, Seyed Abbas Hosseini, Ehsan Pajouheshgar, Farnam Mansouri, and Hamid R Rabiee (2020). « ChOracle: A unified statistical framework for churn prediction ». In: *IEEE Transactions on Knowledge and Data Engineering* 34.4, pp. 1656–1666.

Kidger, Patrick, Patric Bonnier, Imanol Perez Arribas, Cristopher Salvi, and Terry Lyons (2019). « Deep signature transforms ». In: *Advances in Neural Information Processing Systems* 32.

Kidger, Patrick and Terry Lyons (2020). « Signatory: differentiable computations of the signature and logsignature transforms, on both CPU and GPU ». In: *International Conference on Learning Representations*.

Kidger, Patrick, James Morrill, James Foster, and Terry Lyons (2020). « Neural controlled differential equations for irregular time series ». In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., pp. 6696–6707.

Kingma, Diederik P and Jimmy Ba (2014). « Adam: A method for stochastic optimization ». In: *arXiv preprint arXiv:1412.6980*.

Kingma, Diederik P and Max Welling (2013). « Auto-encoding variational bayes ». In: *arXiv preprint arXiv:1312.6114*.

Klein, John P (1992). « Semiparametric estimation of random effects using the Cox model based on the EM algorithm ». In: *Biometrics*, pp. 795–806.

Klein, John P and Melvin L Moeschberger (2005). *Survival analysis: techniques for censored and truncated data*. New York: Springer Science & Business Media.

Kuiper, Edith MM, Bettina E Hansen, Richard A de Vries, Jannie W den Ouden–Muller, Theo JM Van Ditzhuijsen, Els B Haagsma, Martin HMG Houben, Ben JM Witteman, Karel J van Erpecum, Henk R van Buuren, et al. (2009). « Improved prognosis of patients with primary biliary cirrhosis that have a biochemical response to ursodeoxycholic acid ». In: *Gastroenterology* 136.4, pp. 1281–1287.

Kvamme, Håvard and Ørnulf Borgan (2021). « Continuous and discrete-time survival prediction with neural networks ». In: *Lifetime data analysis* 27.4, pp. 710–736.

— (2023). « The Brier Score under Administrative Censoring: Problems and a Solution ». In: *Journal of Machine Learning Research* 24.2, pp. 1–26.

Kvamme, Håvard, Ørnulf Borgan, and Ida Scheel (2019). « Time-to-event prediction with neural networks and Cox regression ». In: *Journal of machine learning research* 20.129, pp. 1–30.

Laird, Nan M and James H Ware (1982). « Random-effects models for longitudinal data ». In: *Biometrics*, pp. 963–974.

Larivière, Bart and Dirk Van den Poel (2004). « Investigating the role of product features in preventing customer churn, by using survival analysis and choice modeling: The case of financial services ». In: *Expert Systems with Applications* 27.2, pp. 277–285.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). « Deep learning ». In: *nature* 521.7553, pp. 436–444.

Lee, Changhee, Jinsung Yoon, and Mihaela Van Der Schaar (2019). « Dynamic-deephit: A deep learning approach for dynamic survival analysis with competing risks based on longitudinal data ». In: *IEEE Transactions on Biomedical Engineering* 67.1, pp. 122–133.

Lee, Changhee, William Zame, Jinsung Yoon, and Mihaela Van Der Schaar (2018). « Deephit: A deep learning approach to survival analysis with competing risks ». In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1.

Lee, Mei-Ling Ting and George A Whitmore (2006). « Threshold regression for survival analysis: modeling event times by a stochastic process reaching a boundary ». In.

Lemler, Sarah (2016). « Oracle inequalities for the Lasso in the high-dimensional Aalen multiplicative intensity model ». In: *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques* 52.2, pp. 981–1008.

Lin, Haiqun, Charles E McCulloch, and Susan T Mayne (2002a). « Maximum likelihood estimation in the joint analysis of time-to-event and multiple longitudinal variables ». In: *Statistics in Medicine* 21.16, pp. 2369–2382.

Lin, Haiqun, Bruce W Turnbull, Charles E McCulloch, and Elizabeth H Slate (2002b). « Latent class models for joint analysis of longitudinal biomarker and event process data: application to longitudinal prostate-specific antigen readings and prostate cancer ». In: *Journal of the American Statistical Association* 97.457, pp. 53–65.

Lin, Ping and Jiongmin Yong (2020). « Controlled singular Volterra integral equations and Pontryagin maximum principle ». In: *SIAM Journal on Control and Optimization* 58.1, pp. 136–164.

Long, Jeffrey D and James A Mills (2018). « Joint modeling of multivariate longitudinal data and survival data in several observational studies of Huntington's disease ». In: *BMC medical research methodology* 18.1, pp. 1–15.

Lyons, Terry and Andrew D McLeod (2022). « Signature Methods in Machine Learning ». In: *arXiv preprint arXiv:2206.14674*.

Lyons, Terry J, Michael Caruana, and Thierry Lévy (2007). *Differential equations driven by rough paths*. Springer.

Marion, Pierre, Adeline Fermanian, Gérard Biau, and Jean-Philippe Vert (2022). « Scaling ResNets in the Large-depth Regime ». In: *arXiv preprint arXiv:2206.06929*.

Masoudnia, Saeed and Reza Ebrahimpour (2014). « Mixture of experts: a literature survey ». In: *Artificial Intelligence Review* 42, pp. 275–293.

McLachlan, Geoffrey J and Thriyambakam Krishnan (2007). *The EM algorithm and extensions*. John Wiley & Sons.

Mei, Hongyuan and Jason M Eisner (2017). « The Neural Hawkes Process: A Neurally Self-Modulating Multivariate Point Process ». In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30.

Mittal, Vikas and Wagner A Kamakura (2001). « Satisfaction, repurchase intent, and repurchase behavior: Investigating the moderating effect of customer characteristics ». In: *Journal of marketing research* 38.1, pp. 131–142.

Molenberghs, Geert, Geert Verbeke, Clarice GB Demétrio, and Afrânio MC Vieira (2010). « A family of generalized linear models for repeated measures with normal and conjugate random effects ». In.

Moon, Intae, Stefan Groha, and Alexander Gusev (2022). « SurvLatent ODE: A Neural ODE based time-to-event model with competing risks for longitudinal data improves cancer-associated Venous Thromboembolism (VTE) prediction ». In: *Machine Learning for Healthcare Conference*. PMLR, pp. 800–827.

Moreau, Jean Jacques (1962). « Fonctions convexes duales et points proximaux dans un espace hilbertien ». In: *Comptes rendus hebdomadaires des séances de l'Académie des sciences* 255, pp. 2897–2899.

Morrill, James, Adeline Fermanian, Patrick Kidger, and Terry Lyons (2020). « A generalised signature method for multivariate time series feature extraction ». In: *arXiv preprint arXiv:2006.00873*.

Morrill, James, Patrick Kidger, Lingyi Yang, and Terry Lyons (2021). « Neural Controlled Differential Equations for Online Prediction Tasks ». In: *arXiv preprint arXiv:2106.11028*.

Mozer, Michael C, Richard Wolniewicz, David B Grimes, Eric Johnson, and Howard Kaushansky (2000). « Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry ». In: *IEEE Transactions on neural networks* 11.3, pp. 690–696.

Mukherjee, B. N. and S. S. Maiti (1988). « On some properties of positive definite Toeplitz matrices and their possible applications ». In: *Linear algebra and its applications* 102, pp. 211–240.

Murray, James and Pete Philipson (2022). « A fast approximate EM algorithm for joint models of survival and multivariate longitudinal data ». In: *Computational Statistics & Data Analysis* 170, p. 107438.

Nagpal, Chirag, Vincent Jeanselme, and Artur Dubrawski (2021). « Deep parametric time-to-event regression with time-varying covariates ». In: *Survival prediction-algorithms, challenges and applications*. PMLR, pp. 184–193.

Nardi, Yuval and Alessandro Rinaldo (Jan. 2008). « On the asymptotic properties of the group Lasso estimator for linear models ». In: *Electronic Journal of Statistics* 0. DOI: *10.1214/08-EJS200*.

Nasejje, Justine B, Henry Mwambi, Keertan Dheda, and Maia Lesosky (2017). « A comparison of the conditional inference survival forest model to random survival forests based on a simulation study as well as on two applications with time-to-event data ». In: *BMC medical research methodology* 17, pp. 1–17.

Nguyen, Van Tuan, Adeline Fermanian, Agathe Guilloux, Antoine Barbieri, Sarah Zohar, Anne-Sophie Jannot, and Simon Bussy (2023). « FLASH: a Fast joint model for Longitudinal And Survival data in High dimension ». In: *arXiv preprint arXiv:2309.03714*.

Ogata, Yosihiko (1988). « Statistical models for earthquake occurrences and residual analysis for point processes ». In: *Journal of the American Statistical association*, pp. 9–27.

Omi, Takahiro, naonori ueda naonori, and Kazuyuki Aihara (2019). « Fully Neural Network based Model for General Temporal Point Processes ». In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32.

Panigutti, Cecilia, Ronan Hamon, Isabelle Hupont, David Fernandez Llorca, Delia Fano Yela, Henrik Junklewitz, Salvatore Scalzo, Gabriele Mazzini, Ignacio Sanchez, Josep Soler Garrido, et al. (2023). « The role of explainable AI in the context of the AI Act ». In: *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1139–1150.

Perez, C Fiorella Murillo, Stephanie Ioannou, Iman Hassanally, Palak J Trivedi, Christophe Corpechot, Adriaan J van der Meer, Willem J Lammers, Pier Maria Battezzati, Keith D Lindor, Frederik Nevens, et al. (2023). « Optimizing therapy in primary biliary cholangitis: Alkaline phosphatase at six months identifies one-year non-responders and predicts survival ». In: *Liver international: official journal of the International Association for the Study of the Liver* 43.7, pp. 1497–1506.

Perez, Carla F Murillo, Maren H Harms, Keith D Lindor, Henk R Van Buuren, Gideon M Hirschfield, Christophe Corpechot, Adriaan J Van Der Meer, Jordan J Feld, Aliya Gulamhusein, Willem J Lammers, et al. (2020). « Goals of treatment for improved survival in primary biliary cholangitis: treatment target should be bilirubin within the normal range and normalization of alkaline phosphatase ». In: *Official journal of the American College of Gastroenterology| ACG* 115.7, pp. 1066–1074.

Periáñez, África, Alain Saas, Anna Guitart, and Colin Magne (2016). « Churn prediction in mobile social games: Towards a complete assessment using survival ensembles ». In:

2016 IEEE international conference on data science and advanced analytics (DSAA). IEEE, pp. 564–573.

Pölsterl, Sebastian (2020). « scikit-survival: A Library for Time-to-Event Analysis Built on Top of scikit-learn ». In: *Journal of Machine Learning Research* 21.212, pp. 1–6.

Proust-Lima, Cécile, Pierre Joly, Jean-François Dartigues, and Hélène Jacqmin-Gadda (2009). « Joint modelling of multivariate longitudinal outcomes and a time-to-event: a nonlinear latent class approach ». In: *Computational statistics & data analysis* 53.4, pp. 1142–1154.

Proust-Lima, Cécile, Viviane Philipps, and Benoit Liquet (2015). « Estimation of extended mixed models using latent classes and latent processes: the R package lcmm ». In: *arXiv preprint arXiv:1503.00890*.

Proust-Lima, Cécile, Mbéry Séne, Jeremy MG Taylor, and Hélène Jacqmin-Gadda (2014). « Joint latent class models for longitudinal and time-to-event data: A review ». In: *Statistical methods in medical research* 23.1, pp. 74–90.

Proust-Lima, Cécile and Jeremy MG Taylor (2009). « Development and validation of a dynamic prognostic tool for prostate cancer recurrence using repeated measures of posttreatment PSA: a joint modeling approach ». In: *Biostatistics* 10.3, pp. 535–549.

Raith, Eamon P, Andrew A Udy, Michael Bailey, Steven McGloughlin, Christopher MacIsaac, Rinaldo Bellomo, David V Pilcher, et al. (2017). « Prognostic accuracy of the SOFA score, SIRS criteria, and qSOFA score for in-hospital mortality among adults with suspected infection admitted to the intensive care unit ». In: *Jama* 317.3, pp. 290–300.

Rasheed, Khansa, Adnan Qayyum, Junaid Qadir, Shobi Sivathamboo, Patrick Kwan, Levin Kuhlmann, Terence O'Brien, and Adeel Razi (2020). « Machine learning for predicting epileptic seizures using EEG signals: A review ». In: *IEEE Reviews in Biomedical Engineering* 14, pp. 139–155.

Reizenstein, Jeremy F. and Benjamin Graham (2020). « Algorithm 1004: The Iisignature Library: Efficient Calculation of Iterated-Integral Signatures and Log Signatures ». In: *ACM Transactions on Mathematical Software* 46.1, pp. 1–21.

Rizopoulos, Dimitris (2012). *Joint models for longitudinal and time-to-event data: With applications in R*. CRC press.

— (2016a). « The R Package JMbayes for Fitting Joint Models for Longitudinal and Time-to-Event Data Using MCMC ». In: *Journal of Statistical Software, Articles* 72.7, pp. 1–46. ISSN: 1548-7660.

— (2016b). « The R Package JMbayes for Fitting Joint Models for Longitudinal and Time-to-Event Data Using MCMC ». In: *Journal of Statistical Software* 72.7, pp. 1–46.

Rizopoulos, Dimitris and Pulak Ghosh (2011). « A Bayesian semiparametric multivariate joint model for multiple longitudinal outcomes and a time-to-event ». In: *Statistics in medicine* 30.12, pp. 1366–1380.

Rubanova, Yulia, Ricky TQ Chen, and David K Duvenaud (2019). « Latent ordinary differential equations for irregularly-sampled time series ». In: *Advances in neural information processing systems* 32.

Rustand, Denis, Janet Van Niekerk, Elias Teixeira Krainski, Håvard Rue, and Cécile Proust-Lima (2024). « Fast and flexible inference for joint models of multivariate longitudinal and survival data using integrated nested Laplace approximations ». In: *Biostatistics* 25.2, pp. 429–448.

Salvi, Cristopher, Maud Lemercier, Chong Liu, Blanka Horvath, Theodoros Damoulas, and Terry Lyons (2021). « Higher Order Kernel Mean Embeddings to Capture Filtrations of Stochastic Processes ». In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34, pp. 16635–16647.

Santis, Rodrigo Barbosa de, Tiago Silveira Gontijo, and Marcelo Azevedo Costa (2022). « A data-driven framework for small hydroelectric plant prognosis using Tsfresh and machine learning survival models ». In: *Sensors* 23.1, p. 12.

Saxena, Abhinav, Kai Goebel, Don Simon, and Neil Eklund (2008). « Damage propagation modeling for aircraft engine run-to-failure simulation ». In: *2008 International Conference on Prognostics and Health Management*, pp. 1–9.

Schemper, Michael and Robin Henderson (2000). « Predictive accuracy and explained variation in Cox regression ». In: *Biometrics* 56.1, pp. 249–255.

Shchur, Oleksandr, Ali Caner Türkmen, Tim Januschowski, and Stephan Günnemann (2021). « Neural temporal point processes: A review ». In: *arXiv preprint arXiv:2104.03528*.

Sifa, Rafet, Christian Bauckhage, and Anders Drachen (2014). « The playtime principle: Large-scale cross-games interest modeling ». In: *2014 IEEE conference on computational intelligence and games*. IEEE, pp. 1–8.

Simeoni, Monica, Paolo Magni, Cristiano Cammia, Giuseppe De Nicolao, Valter Croci, Enrico Pesenti, Massimiliano Germani, Italo Poggesi, and Maurizio Rocchetti (2004). « Predictive pharmacokinetic-pharmacodynamic modeling of tumor growth kinetics in xenograft models after administration of anticancer agents ». In: *Cancer research* 64.3, pp. 1094–1101.

Simon, Noah, Jerome Friedman, Trevor Hastie, and Robert Tibshirani (2013). « A sparse-group lasso ». In: *Journal of Computational and Graphical Statistics* 22.2, pp. 231–245.

Sweeting, Michael J, Jessica K Barrett, Simon G Thompson, and Angela M Wood (2017). « The use of repeated blood pressure measures for cardiovascular risk prediction: a comparison of statistical models in the ARIC study ». In: *Statistics in medicine* 36.28, pp. 4514–4528.

Tang, Weijing, Jiaqi Ma, Qiaozhu Mei, and Ji Zhu (2022). « SODEN: A Scalable Continuous-Time Survival Model through Ordinary Differential Equation Networks ». In: *Journal of Machine Learning Research* 23.34, pp. 1–29.

Therneau, Terry M. and Patricia M. Grambsch (2000). *Modeling survival data: extending the Cox model*. New-York: Springer.

Tong, Jianyang and Xuejing Zhao (2022). « Deep survival algorithm based on nuclear norm ». In: *Journal of Statistical Computation and Simulation* 92.9, pp. 1964–1976.

Tsiatis, Anastasios A and Marie Davidian (2004). « Joint modeling of longitudinal and time-to-event data: an overview ». In: *Statistica Sinica*, pp. 809–834.

Tutz, Gerhard, Matthias Schmid, et al. (2016). *Modeling discrete time-to-event data*. Springer.

Uno, Hajime, Tianxi Cai, Michael J Pencina, Ralph B D'Agostino, and Lee-Jen Wei (2011). « On the C-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data ». In: *Statistics in medicine* 30.10, pp. 1105–1117.

Van Houwelingen, Hans C (2007). « Dynamic prediction by landmarking in event history analysis ». In: *Scandinavian Journal of Statistics* 34.1, pp. 70–85.

Vanderschueren, Toon, Alicia Curth, Wouter Verbeke, and Mihaela Van Der Schaar (2023). « Accounting For Informative Sampling When Learning to Forecast Treatment Outcomes Over Time ». In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett. Vol. 202. PMLR, pp. 34855–34874.

Verbeke, Wouter, Karel Dejaeger, David Martens, Joon Hur, and Bart Baesens (2012). « New insights into churn prediction in the telecommunication sector: A profit driven data mining approach ». In: *European journal of operational research* 218.1, pp. 211–229.

Virmaux, Aladin and Kevin Scaman (2018). « Lipschitz regularity of deep neural networks: analysis and efficient estimation ». In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31.

Virtanen, Pauli, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. (2020). « SciPy 1.0: fundamental algorithms for scientific computing in Python ». In: *Nature methods* 17.3, pp. 261–272.

Wang, Ping, Wei Shen, and Mark Ernest Boye (2012). « Joint modeling of longitudinal outcomes and survival using latent growth modeling approach in a mesothelioma trial ». In: *Health Services and Outcomes Research Methodology* 12.2-3, pp. 182–199.

Wei, Chih-Ping and I-Tang Chiu (2002). « Turning telecommunications call details to churn prediction: a data mining approach ». In: *Expert systems with applications* 23.2, pp. 103–112.

Wiegrebe, Simon, Philipp Kopper, Raphael Sonabend, Bernd Bischl, and Andreas Bender (2024). « Deep learning for survival analysis: a review ». In: *Artificial Intelligence Review* 57.3, p. 65.

Wood, Simon N (2017). *Generalized additive models: an introduction with R*. chapman and hall/CRC.

Wulfsohn, Michael S and Anastasios A Tsiatis (1997). « A joint model for survival and longitudinal data measured with error ». In: *Biometrics*, pp. 330–339.

Xu, Jane and Scott L Zeger (2001). « The evaluation of multiple surrogate endpoints ». In: *Biometrics* 57.1, pp. 81–87.

Yang, Zhongguo, Irshad Ahmed Abbasi, Elfatih Elmubarak Mustafa, Sikandar Ali, and Mingzhu Zhang (2021). « An anomaly detection algorithm selection service for IoT stream data based on tsfresh tool and genetic algorithm ». In: *Security and Communication Networks* 2021.1, p. 6677027.

Yousefi, Safoora, Fatemeh Amrollahi, Mohamed Amgad, Chengliang Dong, Joshua E Lewis, Congzheng Song, David A Gutman, Sameer H Halani, Jose Enrique Velazquez Vega, Daniel J Brat, et al. (2017). « Predicting clinical outcomes from large scale cancer genomic profiles with deep survival models ». In: *Scientific reports* 7.1, pp. 1–11.

Yu, Chun-Nam, Russell Greiner, Hsiu-Chin Lin, and Vickie Baracos (2011). « Learning patient-specific cancer survival distributions as a sequence of dependent regressors ». In: *Advances in neural information processing systems* 24.

Yu, Menggang, Ngayee J Law, Jeremy MG Taylor, and Howard M Sandler (2004). « Joint longitudinal-survival-cure models and their application to prostate cancer ». In: *Statistica Sinica*, pp. 835–862.

Yuan, Lei, Jun Liu, and Jieping Ye (2011). « Efficient methods for overlapping group lasso ». In: *Advances in neural information processing systems* 24, pp. 352–360.

Zhang, Rong, Weiping Li, Wei Tan, and Tong Mo (2017). « Deep and shallow model for insurance churn prediction service ». In: *2017 IEEE International Conference on Services Computing (SCC)*. IEEE, pp. 346–353.

Zhang, Zhongheng, Jaakko Reinikainen, Kazeem Adedayo Adeleke, Marcel E Pieterse, and Catharina GM Groothuis-Oudshoorn (2018). « Time-varying covariates and coefficients in Cox regression models ». In: *Annals of Translational Medicine* 6.7.

Zhu, Ciyou, Richard H Byrd, Peihuang Lu, and Jorge Nocedal (1997). « Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization ». In: *ACM Transactions on Mathematical Software (TOMS)* 23.4, pp. 550–560.

Zou, Hui and Trevor Hastie (2005). « Regularization and variable selection via the elastic net ». In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.2, pp. 301–320.