Lights: a generalized joint model for high-dimensional multivariate longitudinal data and censored durations

Simon Bussy*[1,2], Van Tuan Nguyen[2], Antoine Barbieri[3], Sarah Zohar[1], and Anne-Sophie Jannot[1,4]

[1]*INSERM, UMRS 1138, Centre de Recherche des Cordeliers, Paris, France*
[2]*Califrais' Machine Learning Lab, Paris, France*
[3]*INSERM, UMR 1219, Bordeaux Population Health Research Center, Univ. Bordeaux, France*
[4]*Biomedical Informatics and Public Health Department, EGPH, APHP, Paris, France*

Abstract

This paper introduces a prognostic method called *lights* to deal with the problem of joint modeling of longitudinal data and censored durations, where a large number of longitudinal features are available. Yet there is no standard model so far to learn from such high-dimensional multivariate longitudinal data in a survival analysis setting. Features are extracted from the longitudinal processes and included as potential risk factor in a group-specific Cox model with high-dimensional shared associations. Appropriate penalties are then used during inference to allow flexibility in modeling the dependency between the longitudinal features and the event time. The statistical performance of the method is examined on an extensive Monte Carlo simulation study, and finally illustrated on publicly available datasets. On these high-dimensional datasets, our proposed method significantly outperforms the state-of-the-art survival models regarding risk prediction in terms of C-index, with a computing time orders of magnitude faster. In addition, it provides powerful interpretability by automatically pinpointing significant features being relevant from a practical perspective. Thus, we propose a powerfull tool with the ability of automatically determining significant prognostic longitudinal features, which is of increasing importance in many areas: for instance personalized medicine, or churn prediction in a customer profile and activity monitoring setting, to name but a few.

Keywords. High-dimensional estimation; Joint modeling; Multivariate longitudinal data; Survival analysis

1 Introduction

With the increasing expectations to know their customers from account opening throughout the duration of the business relationship, web companies have the luxury of building elaborate systems to help them keep everything on track. The amount of recorded data per client is often tremendous and growing through time. There is no tool today to take into account simultaneously a huge number of longitudinal signals in a high-dimensional context to perform real-time churn (or satisfaction) risk prediction. Similarly, in many clinical studies, it has become increasingly common to record the values of longitudinal features (e.g., biomarkers) until the occurrence of an event of interest for a subject. The "joint modeling" approaches, namely modeling the longitudinal and survival outcomes

*Corresponding author: simon.bussy@gmail.com

through a joint likelihood model rather than separately, has received considerable attention during the past two decades [**?**]. Numerical studies suggest that these approaches are among the most satisfactory to combine information [**?**]. They have the advantage of making more efficient use of the data since information about survival also goes into modeling the longitudinal features. In addition, they produce unbiased estimates and do not rely on approximations for incorporating conplex longitudinal trajectories. Most developments have either focused on shared random-effect models (SREMs) [**?**], in which characteristics of the longitudinal processes (for instance functions of the random effects) are included as features in the survival model ; or on joint latent class models (JLCMs) [**?**], in which the population is considered as heterogeneous, with the assumption that there exist homogeneous latent subgroups sharing the same marker trajectories and the same prognostic.

The high-dimensional longitudinal data context. With the exploding number of daily internet users, or with the development of electronic health records in a medical context, high-dimensional settings are becoming increasingly frequent in various contexts where the number of available features to consider as potential risk factors is tremendous. Moreover, with an increased focus on personalised medicine, the need to implement multivariate models that account for a large number of longitudinal outcomes is critical. Despite this, joint models have predominantly focused on univariate data, with attempts to fit multiple univariate joint models separately [**?**], which is inefficient [**?**]. Despite many multivariate models being presented in full generality, questions arrising from the high-dimensional context – e.g., computational power, limits in numerical estimation, or sample size – are never considered in analyses (to the best of our knowledge), and the number of longitudinal outcomes considered in numerical studies are often very low (see **?** for a complete review). For instance, **?** only considers 3 longitudinal outcomes in the simulation study while mentioning a "high-dimensional multivariate longitudinal data" context.

General framework. The setting of this paper is such that we want to incorporate high-dimensional time-dependent (longitudinal) features measured with error in a survival model. Let us consider the usual survival analysis framework. Following **?**, let non-negative random variables $T^\star$ and C stand for the times of the event of interest and censoring times respectively. The event of interest could be for instance survival time, re-hospitalization, relapse or disease progression in a medical context; or the time when a client stops using a company's product or service in a churn prediction setting. We then denote T the right-censored time and Δ the censoring indicator, defined as

$$T = T^\star \wedge C \quad \text{and} \quad \Delta = \mathbb{1}_{\{T^\star \leq C\}}$$

respectively, where $a \wedge b$ denotes the minimum between two numbers a and b, and $\mathbb{1}_{\{\cdot\}}$ the indicator function taking the value 1 if the condition in $\{\cdot\}$ is satisfied and 0 otherwise.

Let X denotes the p-dimensional vector of time-independent features (e.g., patients characteristics, therapeutic strategy, or omics features recorded at the begining of a medical study), and let $Y(t) = \left(Y^1(t), \ldots, Y^L(t)\right)^\top \in \mathbb{R}^L$ denote the value of the L-dimensional longitudinal outcome at time point $t \geq 0$, with $L \in \mathbb{N}_+$.

Heterogeneity of the population. An assumption of heterogeneity within the subject population is frequently relevant in medical research where several differing profiles of subjects are expected [**?**]. To take account of this, we introduce a latent variable $G \in$

$\{0, \ldots, K-1\}$ modeling the $K \geq 1$ subgroups of different risk, which is a classical modeling assumption in JLCMs [**??**]. Let us denote

$$\pi_{\xi_k}(x) = \mathbb{P}[G = k | X = x] \tag{1}$$

the latent class membership probability given time-independent features $x \in \mathbb{R}^p$, and consider a softmax link function given by

$$\pi_{\xi_k}(x) = \frac{e^{x^\top \xi_k}}{\sum_{k=0}^{K-1} e^{x^\top \xi_k}} \tag{2}$$

where $\xi_k \in \mathbb{R}^p$ denotes a vector of coefficients that quantifies the impact of each time-independent features on the probability that a subject belongs to the k-th group, with $\xi_0 = \mathbf{0}_p$ for overparameterization purpose, where $\mathbf{0}_p$ stands for the vector of $\mathbb{R}^p$ having all coordinates equal to zero. The intercept term is here omitted without loss of generality. From now on, all computations are done conditionally on features x.

Main contribution. In this paper, we propose a method called *lights* (generaLized joInt hiGH-dimensional longiTudinal Survival) which is from both JLCMs and SREMs, since we also include features extracted from the longitudinal processes as potential risk factor in the survival model, which is a group-specific Cox model [**?**] with high-dimensional shared associations. To allow flexibility in modeling the dependency between the longitudinal features and the event time, we use appropriate penalties : elastic net [**?**] for feature selection in the latent class membership, and sparse group lasso [**?**] in the survival model, as well as for the fixed effect (allowing flexible representations of time). Indeed, this penalty acts like the lasso at the trajectory level, namely an entire trajectory may drop out of the model on one side. On the other hand, it yields sparsity for a given trajectory, namely feature selection. Inference is achieved using an efficient and novel Quasi-Newton Monte Carlo Expectation Maximization algorithm. Posterior estimates of the latent class membership probabilities given the longitudinal process and the fact that the subject is still alive at prediction time – in a so called "real-time prediction paradigm" – are then used as discriminative marker rule for risk prediction in the cross-validation procedure for selecting the best regularization hyper-parameters (to perform feature selection and adapt to the longitudinal data complexity). Hence, the method provides interpretations of the high-dimensional longitudinal features, thus offering a powerful tool for clinical decision making in patient monitoring for instance, or else for real-time decision support in a customer's satisfaction monitoring context.

Organization of the paper. A precise description of the model is given in Section 2. Section 3 focuses on a regularized version of the model to exploit dimension reduction and prevent overfitting. Inference is presented under this framework, as well as the developed algorithm. Section 4 introduces the C-index metric, as well as a novel evaluation strategy to assess diagnostic prediction performances while mimicking a real-time use of the model in clinical care, and finally the considered competing methods. Section 5 presents the simulation procedure used to evaluate the performance of our method in a high-dimensional context and compares it with state-of-the-art ones. In Section 6, we apply our method to high-dimensional publicly available datasets. Finally, we discuss the obtained results in Section 7.

Notations. Throughout the paper, for every $q > 0$, we denote by $\|v\|_q$ the usual ℓ_q-quasi norm of a vector $v \in \mathbb{R}^m$, namely $\|v\|_q = (\sum_{k=1}^m |v_k|^q)^{1/q}$. We also denote $\|v\|_0 = |\{k : v_k \neq 0\}|$, where $|A|$ stands for the cardinality of a finite set A. $\lfloor a \rfloor$ denotes the largest integer less than or equal to $a \in \mathbb{R}$. For $u, v \in \mathbb{R}^m$, we denote by $u \odot v$ the Hadamard product $u \odot v = (u_1 v_1, \ldots, u_m v_m)^\top$. For a squared matrix M, vech(M) stacks columns of M one under another in a single vector, starting each column at its diagonal element. We write I_m for the identity matrix of $\mathbb{R}^{m \times m}$. Finally, we write, for short, $\mathbf{1}_m$ (resp. $\mathbf{0}_m$) for the vector of $\mathbb{R}^m$ having all coordinates equal to one (resp. zero).

2 Method

In this section, we descibe the longitudinal and time-to-event submodels, as well as the required hypothesis in order to write a likelihood and draw inference for the lights model.

2.1 Group-specific marker trajectories

We suppose a group-specific marker trajectory and a generalized linear mixed model for each longitudinal marker given subgroup G, so that for the l-th outcome at time $t \geq 0$ one has

$$h_l\big(\mathbb{E}[Y^l(t)|b^l, G = k]\big) = m_k^l(t) \tag{3}$$

where h_l denotes a known one-to-one link function, and m_k^l the linear predictor such that

$$m_k^l(t) = u^l(t)^\top \beta_k^l + v^l(t)^\top b^l$$

where $u^l(t) \in \mathbb{R}^{q_l}$ is a row vector of (possibly) time-varying features with corresponding unknown fixed effect parameters β_k^l, and $v^l(t) \in \mathbb{R}^{r_l}$ is a row vector of (possibly) time-varying features with $r_l \leq q_l$ and where the corresponding subject-and-longitudinal outcome specific random effects b^l that does not depend on the group membership, which is not a strong modeling assumption.

Assumption 1. *We suppose that the random effects are independent of the group membership, and that the latter remain independent conditional on the observed data (namely T, Δ and Y).*

A suitable distributional assumption for the random effects component is a zero-mean multivariate normal distribution [**?**], that is

$$b^l \sim \mathcal{N}(0, D_{ll}) \tag{4}$$

with $D_{ll} \in \mathbb{R}^{r_l \times r_l}$ the unstructured variance-covariance matrix. To account for dependence between the different longitudinal outcome types, we let $\mathrm{Cov}[b^l, b^{l'}] = D_{ll'}$ for $l \neq l'$ and we denote

$$D = \begin{bmatrix} D_{11} & \cdots & D_{1L} \\ \vdots & \ddots & \vdots \\ D_{1L}^\top & \cdots & D_{LL} \end{bmatrix}$$

the global variance-covariance matrix. In the sequel of the paper, our aim is to associate the true and unobserved value $m_k^l(t)$ of the l-th longitudinal outcome at time t with the event outcome $T^\star$.

2.2 Group-specific risk of event

To quantify the effect of the longitudinal outcomes on the risk for an event, we use a Cox [**?**] relative risk model of the form

$$\lambda(t|\mathcal{M}_k(t), G=k) = \lambda_0(t)\exp\left\{x^\top\gamma_k^0 + \sum_{l=1}^{L}\sum_{a=1}^{\mathcal{A}} {\gamma_{k,a}^l}^\top \varphi_a(t,\beta_k^l,b^l)\right\}, \tag{5}$$

where $\lambda_0(\cdot)$ is an unspecified baseline hazard function and given that $G=k$, we denote

$$\mathcal{M}_k(t) = \{m_k^l(u), 0 \leq u < t\}$$

the history of the true unobserved longitudinal process up to time t. We choose to incorporate x, also used in (1), with no *a priori* on the choice of time-independent features involved in the definition of π_{ξ_k} nor λ, since independent regularizations are used during inference, see (9). Corresponding fixed effects are denoted $\gamma_k^0 \in \mathbb{R}^p$, and for each l-th longitudinal outcome, we consider $\mathcal{A} \in \mathbb{N}_+$ known functionals φ_a defining a shared association with $\gamma_{k,a}^l \in \mathbb{R}^{\imath_a}$ the corresponding joint association parameters, and $\imath_a \in \mathbb{N}_+$ the dimension of the corresponding $\mathrm{Im}(\varphi_a)$. This can be viewed as a generalization of SREMs [**?**]. Let us finally denote

$$({\gamma_k^0}^\top, \gamma_k^\top)^\top = ({\gamma_k^0}^\top, \gamma_{k,1}^1, \ldots, \gamma_{k,\mathcal{A}}^1, \ldots, \gamma_{k,1}^L, \ldots, \gamma_{k,\mathcal{A}}^L)^\top \in \mathbb{R}^{p+L\mathcal{A}}.$$

Specification of the functionals $(\varphi_a)_{a\in\mathcal{A}}$. The association structure between the longitudinal and the time-to-event submodels is key to the joint modeling framework. In spite of that, rationale for selecting shared associations has received little attention. We then propose to include some of the most common parameterization with no *a priori*, set out in Table 1, and let the model select the relevant ones through the regularization strategy described in Section 3.1.

Description	$\varphi_a(t,\beta_k^l,b^l)$	$\frac{\partial\varphi_a(t,\beta_k^l,b^l)}{\partial\beta_k^l}$	$\imath_a$	Reference
Linear predictor	$m_k^l(t)$	$u^l(t)$	1	**?**
Random effects	b^l	$\mathbf{0}_{q_l}$	r_l	**?**
Time-dependent slope	$\frac{\mathrm{d}}{\mathrm{d}t}m_k^l(t)$	$\frac{\mathrm{d}}{\mathrm{d}t}u^l(t)$	1	**?**
Cumulative effect	$\int_0^t m_k^l(s)\mathrm{d}s$	$\int_0^t u^l(s)\mathrm{d}s$	1	**?**

Table 1: Description of the shared associations included in the group-specific risk of event submodel (5). The gradient with respect to β_k^l is also given, which will be useful for inference in Section 3.2.

2.3 Likelihood

Consider an independent and identically distributed (i.i.d.) cohort of n subjects

$$\mathcal{D}_n = \left\{(x_1, y_1^1, \ldots, y_1^L, t_1, \delta_1), \ldots, (x_n, y_n^1, \ldots, y_n^L, t_n, \delta_n)\right\}$$

where for each subject $i = 1, \ldots, n$, process Y_i^l is measured n_i^l times at $t_{i1}^l, \ldots, t_{in_i^l}^l$ (which can differ between subjects and outcomes) with $t_{ij}^l \leq t_{ij+1}^l$ for all $j = 1, \ldots, n_i^l - 1$ and

such that

$$y_i^l = (y_{i1}^l, \dots, y_{in_i^l}^l)^\top \in \mathbb{R}^{n_i^l} \quad \text{with} \quad y_{ij}^l = Y_i^l(t_{ij}^l)$$

for all $l = 1, \dots, L$. For the i-th subject, let us denote

$$\begin{cases} y_i &= ({y_i^1}^\top \cdots {y_i^L}^\top)^\top \in \mathbb{R}^{n_i}, \\ b_i &= ({b_i^1}^\top \cdots {b_i^L}^\top)^\top \in \mathbb{R}^{r}, \end{cases}$$

with $n_i = \sum_{l=1}^L n_i^l$ and $r = \sum_{l=1}^L r_l$ the total number of longitudinal measurements (for subject i) and the total dimension of the random effects respectively, as well as the following design matrices

$$U_i = \begin{bmatrix} U_{i1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & U_{iL} \end{bmatrix} \in \mathbb{R}^{n_i \times q} \quad \text{and} \quad V_i = \begin{bmatrix} V_{i1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & V_{iL} \end{bmatrix} \in \mathbb{R}^{n_i \times r}$$

with $q = \sum_{l=1}^L q_l$ and where for all $l = 1, \dots, L$, one writes

$$\begin{cases} U_{il} &= \left(u_i^l(t_{i1}^l)^\top \cdots u_i^l(t_{in_i^l}^l)^\top\right)^\top \in \mathbb{R}^{n_i^l \times q_l}, \\ V_{il} &= \left(v_i^l(t_{i1}^l)^\top \cdots v_i^l(t_{in_i^l}^l)^\top\right)^\top \in \mathbb{R}^{n_i^l \times r_l}. \end{cases}$$

From now on, all computations are done conditionally on the design matrices $(U_i)_{i=1,\dots,n}$ and $(V_i)_{i=1,\dots,n}$.

Assumption 2. *Suppose that for a given subject i and conditional on the random effects and the group membership, all Y_i^l are independent for $l = 1, \dots, L$.*

Assumption 2 is similar to standard modeling hypothesis, see for instance **?**. For all $k = 0, \dots, K-1$, we denote

$$\beta_k = ({\beta_k^1}^\top \cdots {\beta_k^L}^\top)^\top \in \mathbb{R}^q$$

and

$$M_{ik} = U_i \beta_k + V_i b_i \in \mathbb{R}^{n_i}.$$

Given (3) and Assumption 2, each y_i^l is assumed to be from a one-parameter exponential family with respect to a reference measure which is either the Lebesgue measure (e.g., in the Gaussian case) or the counting measure (e.g., in the logistic cases). The conditional distribution of $y_i | b_i, G_i = k$ is then assumed to be from a distribution with a density of the form

$$f(y_i | b_i, G_i = k) = \exp\left\{(y_i \odot \Phi_i)^\top M_{ik} - c_\phi(M_{ik}) + d_\phi(y_i)\right\}, \tag{6}$$

with

$$\Phi_i = (\phi_1^{-1} {\mathbf{1}_{n_i^1}}^\top \cdots \phi_L^{-1} {\mathbf{1}_{n_i^L}}^\top)^\top \in \mathbb{R}^{n_i}$$

and $\phi = (\phi_1, \dots, \phi_L)^\top \in \mathbb{R}^L$. The density described in (6) encompasses several distributions, see Table 2. The functions $c_\phi(\cdot)$ and $d_\phi(\cdot)$ are known as well as the dispersion parameters ϕ_l, while parameters β_k have to be estimated. Note that $d_\phi(\cdot)$ is related to the normalizing constant.

Let us denote

$$\theta = \left(\xi_0^\top \cdots \xi_{K-1}^\top, \beta_0^\top \cdots \beta_{K-1}^\top, \phi^\top, \mathrm{vech}(D), \lambda_0, \gamma_0^\top \cdots \gamma_{K-1}^\top\right)^\top \in \mathbb{R}^\vartheta$$

Model	Support	Use cases	ϕ_l	$h_l(\cdot)$ in (3)	$c_\phi(\cdot)$
Gaussian	$\mathbb{R}$	Continuous response data	σ_l^2	$z \mapsto z$	$z \mapsto z^2/2\sigma_l^2$
Categorical	$\{0,1\}^C$	Outcome with C modalities	1	$z \mapsto \log\left(\frac{z}{1-z}\right)$	$z \mapsto \log(1+e^z)$
Poisson	$\mathbb{N}$	Count of occurrences	1	$z \mapsto \log(z)$	$z \mapsto e^z$

Table 2: Examples of standard distributions that fit in the considered setting, given in the univariate case for simplicity of the notations.

the collection of the $\vartheta \in \mathbb{N}_+$ unknown parameters to estimate. Note that for ease of notations, we include λ_0 in θ although it is a function. To write the log-likelihood $\ell_n(\theta)$ (rescaled by n^{-1}) for samples in $\mathcal{D}_n$, corresponding to the joint distribution of the time-to-event and longitudinal outcomes, let us make the following hypothesis.

Assumption 3. *Assume that both the random effects vector b_i and the group membership account for the association between the longitudinal and event outcomes, that is*

$$f(t_i, \delta_i, y_i | b_i, G_i = k; \theta) = f(t_i, \delta_i | b_i, G_i = k; \theta) f(y_i | b_i, G_i = k; \theta) \tag{7}$$

for all $i = 1, \dots, n$.

Assumption 3 is a generalization of classical hypothesis used in SREMs and JLCMs [**?**]. Then, one has

$$\begin{aligned} \ell_n(\theta) &= \ell_n(\theta; \mathcal{D}_n) \\ &= n^{-1} \sum_{i=1}^{n} \log \sum_{k=0}^{K-1} \pi_{\xi_k}(x_i) f(y_i | G_i = k) \int_{\mathbb{R}^r} f(t_i, \delta_i | b_i, G_i = k; \theta) f(b_i | y_i, G_i = k; \theta) \mathrm{d}b_i, \end{aligned} \tag{8}$$

where

$$f(t_i, \delta_i | b_i, G_i = k; \theta) = \big[\lambda(t_i | \mathcal{M}_k(t_i), G_i = k)\big]^{\delta_i} \exp\Big\{ - \int_0^{t_i} \lambda(s | \mathcal{M}_k(s), G_i = k) \mathrm{d}s \Big\}.$$

Specification of the design matrices. In many practical applications, subjects show highly nonlinear longitudinal trajectories. In (8), the complete longitudinal history is required for the computation of the survival function. Hence, in order to produce a good estimate of $\mathcal{M}_k(t)$, we consider a flexible representations for $u^l(t)$ using a high-dimensional vector of time monomials, namely

$$u^l(t) = (1, t, t^2, \dots, t^\alpha)^\top$$

with $\alpha \in \mathbb{N}_+$. The idea here is to allow a wide range of polynomial orders for the representation so that a suitable one can be automatically chosen for each trajectory – depending on its inherent complexity – thanks to the regularization strategy proposed in (9). We then let

$$v^l(t) = (1, t)^\top$$

so that each trajectory of each subject gets an affine random effect. Hence with this choice in practice, one has $q_l = \alpha + 1$ and $r_l = 2$ for all $l = 1, \dots, L$.

3 Inference

In this section, we describe the procedure for estimating the parameters of the lights model. Let us first present the considered penalized objective, and then focus on the algorithm proposed for inference.

3.1 Penalized objective

In order to avoid overfitting and improve the prediction power of our method, we propose to minimize the penalized objective

$$\ell_n^{\text{pen}}(\theta) = -\ell_n(\theta) + \sum_{k=0}^{K-1} \zeta_{1,k}\big(\|\xi_k\|_{\text{en},\eta} + \|\gamma_k^0\|_{\text{en},\eta}\big) + \zeta_{2,k}\|\beta_k\|_{\text{sgl}_1,\tilde{\eta}} + \zeta_{3,k}\|\gamma_k\|_{\text{sgl}_1,\tilde{\eta}} \tag{9}$$

where for all $k = 0, \ldots, K-1$, we add an elastic net regularization [**?**] of the vector ξ_k and a sparse group lasso regularization [**?**] of the vectors γ_k and β_k, for tuning hyper-parameters $(\zeta_{1,k}, \zeta_{2,k}, \zeta_{3,k})^\top \in \mathbb{R}_+^3$. Here, $(\eta, \tilde{\eta}) \in [0,1]^2$ are fixed and we denote

$$\|z\|_{\text{en},\eta} = (1-\eta)\|z\|_1 + \frac{\eta}{2}\|z\|_2^2$$

for any vector z, that is a linear combination of the lasso (ℓ_1) and ridge (squared ℓ_2) penalties, and

$$\|z\|_{\text{sgl}_1,\tilde{\eta}} = (1-\tilde{\eta})\|z\|_1 + \tilde{\eta}\sum_{l=1}^{L}\|z\|_{2,l}$$

for the sparse group lasso penalty with $\|z\|_{2,l} = \|z^l\|_2$, for which we do not have to account for group sizes since they all have the same one. Hence, the resulting optimization problem is written

$$\hat{\theta} \in \text{argmin}_{\theta\in\mathbb{R}^\vartheta}\, \ell_n^{\text{pen}}(\theta). \tag{10}$$

One advantage of the considered regularization method is its ability to perform feature selection (the lasso part) and pinpoint the most important features relatively to the prediction objective : the support of ξ_k thus informs on the time-independent features involved in the k-th group membership. The ridge part allows to handle potential correlation between features. On the other hand, the sparse group lasso allows to perform feature selection for each trajectory : the support of γ_k^l informs on the features involved in the k-th group risk of event for the l-th longitudinal outcome. Finally, the sparse group lasso of β_k allows to consider a flexible representation of time for the design matrices $(U_i)_{i=1,\ldots,n}$ and lets the model automatically fit each trajectory l with the right complexity. Note that in practice, the intercept is not regularized.

3.2 A Quasi-Newton Monte Carlo EM

In order to derive an algorithm for this objective, we introduce a so-called QNMCEM algorithm, being a combination between an EM algorithm [**?**] with Monte Carlo approximations [**?**], and multiple L-BFGS-B algorithms [**?**]. EM algorithm has already been used for multivariate data joint modeling (see **?** for instance), but here we face different original problems: for each subject i, the latent variables are the pairs (G_i, b_i) (not only the random effects); and then, we want to minimize the penalized objective ℓ_n^{pen} (not "only" the negative log-likelihood).

We first need to compute the negative completed log-likelihood (here scaled by n^{-1}), namely the negative joint distribution of $\mathcal{D}_n$, $\boldsymbol{b} = (b_1, \ldots, b_n)$ and $\boldsymbol{G} = (G_1, \ldots, G_n)$. It can be written

$$\begin{aligned}
\ell_n^{\text{comp}}(\theta) &= \ell_n^{\text{comp}}(\theta; \mathcal{D}_n, \boldsymbol{b}, \boldsymbol{G}) \\
&= -n^{-1} \sum_{i=1}^{n} -\frac{1}{2}(r \log 2\pi + \log |D| + b_i^\top D^{-1} b_i) + \sum_{k=0}^{K-1} \mathbb{1}_{\{G_i=k\}} \Big[\log \pi_{\xi_k}(x_i) \\
&\quad + \delta_i \Big(\log \lambda_0(t_i) + x_i^\top \gamma_k^0 + \sum_{l=1}^{L} \sum_{a=1}^{\mathcal{A}} {\gamma_{k,a}^l}^\top \varphi_a(t_i, \beta_k^l, b_i^l) \Big) \\
&\quad - \int_0^{t_i} \lambda_0(s) \exp \Big\{ x_i^\top \gamma_k^0 + \sum_{l=1}^{L} \sum_{a=1}^{\mathcal{A}} {\gamma_{k,a}^l}^\top \varphi_a(s, \beta_k^l, b_i^l) \Big\} \mathrm{d}s \\
&\quad + (y_i \odot \Phi_i)^\top M_{ik} - c_\phi(M_{ik}) + d_\phi(y_i) \Big].
\end{aligned}$$

Suppose that we are at step $w+1$ of the algorithm, with current iterate denoted $\theta^{(w)}$.

Monte Carlo E-step. We need to compute the expected negative log-likelihood of the complete data conditional on the observed data and the current estimate of the parameters given by

$$\mathcal{Q}_n(\theta, \theta^{(w)}) = \mathbb{E}_{\theta^{(w)}}[\ell_n^{\text{comp}}(\theta) | \mathcal{D}_n].$$

Given Assumption 1, the previous expression requires to compute expectations of the form

$$\mathbb{E}_{\theta^{(w)}}[g_i(b_i, G_i) | t_i, \delta_i, y_i] = \sum_{k=0}^{K-1} \pi_{ik}^{\theta^{(w)}} \int_{\mathbb{R}^r} g(b_i, G_i) f(b_i | t_i, \delta_i, y_i; \theta^{(w)}) \mathrm{d}b_i$$

for different functions g_i, where we denote

$$\pi_{ik}^{\theta^{(w)}} = \mathbb{P}_{\theta^{(w)}}[G_i = k | t_i, \delta_i, y_i] \tag{11}$$

the posterior probability of the latent class membership using parameters $\theta^{(w)}$. Then, either $g_i(b_i, G_i) = \tilde{g}_i(b_i)$ (for instance $\tilde{g}_i : b_i \mapsto b_i^\top D^{-1} b_i$), or $g_i(b_i, G_i) = g_k(G_i)$ with $g_k : G_i \mapsto \mathbb{1}_{\{G_i=k\}}$.

In the case $g_i(b_i, G_i) = \tilde{g}_i(b_i)$, one has

$$\mathbb{E}_{\theta^{(w)}}[\tilde{g}_i(b_i) | t_i, \delta_i, y_i] = \int_{\mathbb{R}^r} \tilde{g}_i(b_i) f(b_i | t_i, \delta_i, y_i; \theta^{(w)}) \mathrm{d}b_i = \frac{\sum_{k=0}^{K-1} \pi_{\xi_k^{(w)}}(x_i) \Lambda_{ik,\tilde{g}_i}^{\theta^{(w)}}}{\sum_{k=0}^{K-1} \pi_{\xi_k^{(w)}}(x_i) \Lambda_{ik,1}^{\theta^{(w)}}} \tag{12}$$

with

$$\begin{aligned}
\Lambda_{ik,\tilde{g}_i}^{\theta^{(w)}} &= \int_{\mathbb{R}^r} \tilde{g}_i(b_i) f(t_i, \delta_i, y_i | b_i, G_i = k; \theta^{(w)}) f(b_i; \theta^{(w)}) \mathrm{d}b_i & (13) \\
&= f(y_i | G_i = k) \int_{\mathbb{R}^r} \tilde{g}_i(b_i) f(t_i, \delta_i | b_i, G_i = k; \theta^{(w)}) f(b_i | y_i, G_i = k; \theta^{(w)}) \mathrm{d}b_i & (14)
\end{aligned}$$

and $\Lambda_{ik,1}^{\theta^{(w)}}$ obtained from $\Lambda_{ik,\tilde{g}_i}^{\theta^{(w)}}$ taking $\tilde{g}_i : b_i \mapsto 1$.

in the Gaussian case used in practice, one can also sample from

$$\begin{aligned}
y_i | G_i = k &\sim \mathcal{N}\big(U_i \beta_k, V_i D V_i^\top + \Sigma_i\big) \\
b_i | y_i, G_i = k &\sim \mathcal{N}\big(W_i V_i^T \Sigma_i^{-1}(y_i - U_i \beta_k), W_i\big)
\end{aligned}$$

with $W_i = (V_i^\top \Sigma_i^{-1} V_i + D^{-1})^{-1}$

The integral in (**??**) is not tractable analytically, and some form of approximation must be used in practice. Standard numerical integration techniques (such as Gaussian quadrature) are not well suited in our high-dimensional context. We then propose to use Monte Carlo approximation, which has already been applied for generalized linear mixed models [**?**], with antithetic normal variates method [**?**] to reduce variance of the simulation result. Algorithm 1 describes how to construct the set $S^{(w)}$ used for the approximation.

Algorithm 1: Construction of the samples in $S_i^{(w)}$

Input: $S_i^{(w)} = \varnothing$

Output: $S_i^{(w)}$ filled with $2N$ samples

Compute $C_i^{(w)} \in \mathbb{R}^{r\times r}$ such that $C_i^{(w)} {C_i^{(w)}}^\top = \mathrm{M}_i$ `// Cholesky decomposition`

for $\breve{n} = 1, \ldots, N$ **do**

  Sample $\Omega_{\breve{n}} \sim \mathcal{N}(0, I_r)$

  Compute $\breve{b}_{\breve{n}} = C_i^{(w)} \Omega_{\breve{n}} \in \mathbb{R}^r$

  Update $S_i^{(w)} \leftarrow S_i^{(w)} \cup \{\breve{b}_{\breve{n}}, -\breve{b}_{\breve{n}}\}$

Return: $S_i^{(w)}$

$\mathrm{M}_i = W_i$ or $\mathrm{M}_i = D^{(w)}$

The approximation of (12) is finally obtained by

$$\hat{\mathbb{E}}_{\theta^{(w)}}[\tilde{g}_i(b_i)|t_i, \delta_i, y_i] = \frac{\sum_{k=0}^{K-1} \pi_{\xi_k^{(w)}}(x_i) \hat{\Lambda}_{ik,\tilde{g}_i}^{\theta^{(w)}}}{\sum_{k=0}^{K-1} \pi_{\xi_k^{(w)}}(x_i) \hat{\Lambda}_{ik,1}^{\theta^{(w)}}} \tag{15}$$

with

$$\hat{\Lambda}_{ik,\tilde{g}_i}^{\theta^{(w)}} = \frac{1}{2N} \sum_{\breve{b}\in S_i^{(w)}} \tilde{g}_i(\breve{b}) f(t_i, \delta_i|\breve{b}, G_i = k; \theta^{(w)}). \tag{16}$$

It is common in Monte Carlo EM to increase N with the iterations w [**?**] since it is computationally inefficient to use a large N in the early steps of the algorithm, when $\theta^{(w)}$ is far from $\hat{\theta}$. Multiple techniques have been proposed (see for instance **?** where a subjectively rule is followed, or **?** that uses a rule based on confidence intervals for $\theta^{(w)}$ which requires additional variance estimation). We opt for a simple automated approach using relative differences of the objective function defined in (9) (see Algorithm 3).

Now in the case $g = g_k$, one has $\mathbb{E}_{\theta^{(w)}}[g_k(Gi)|t_i, \delta_i, y_i] = \pi_{ik}^{\theta^{(w)}}$ and we compute its approximation given

$$\hat{\pi}_{ik}^{\theta^{(w)}} = \frac{\pi_{\xi_k^{(w)}}(x_i) \hat{\Lambda}_{ik,1}^{\theta^{(w)}}}{\sum_{k=0}^{K-1} \pi_{\xi_k^{(w)}}(x_i) \hat{\Lambda}_{ik,1}^{\theta^{(w)}}}. \tag{17}$$

Quasi-Newton M-step. Here, we need to compute

$$\theta^{(w+1)} \in \mathrm{argmin}_{\theta\in\mathbb{R}^\vartheta}\, \mathcal{Q}_n(\theta, \theta^{(w)}) + \sum_{k=0}^{K-1} \zeta_{1,k}\big(\|\xi_k\|_{\mathrm{en},\eta} + \|\gamma_k^0\|_{\mathrm{en},\eta}\big) + \zeta_{2,k}\|\beta_k\|_{\mathrm{sgl}_1,\tilde{\eta}} + \zeta_{3,k}\|\gamma_k\|_{\mathrm{sgl}_1,\tilde{\eta}}.$$

Let us introduce the following useful notations

$$\begin{cases} \tilde{g}^1_{i,s,\gamma_k^0,\gamma_k,\beta_k} & : b_i \mapsto \exp\left\{x_i^\top \gamma_k^0 + \tilde{g}^2_{s,\gamma_k,\beta_k}(b_i)\right\}, \\ \tilde{g}^2_{s,\gamma_k,\beta_k} & : b_i \mapsto \sum_{l=1}^{L} \sum_{a=1}^{\mathcal{A}} {\gamma^l_{k,a}}^\top \varphi_a(s, \beta_k^l, b_i^l). \end{cases} \tag{18}$$

We then present the parameters updates in the order given in Algorithm 3. First, the update of $D^{(w)}$ is naturally given in closed-form by

$$D^{(w+1)} = n^{-1} \sum_{i=1}^{n} \hat{\mathbb{E}}_{\theta^{(w)}}[b_i b_i^\top | t_i, \delta_i, y_i]. \tag{19}$$

Then, let us focus on the update of $\xi_k^{(w)}$ for $k = 0, \ldots, K-1$. We denote

$$P^{(w)}_{n,k}(\xi_k) = -n^{-1} \sum_{i=1}^{n} \hat{\pi}_{ik}^{\theta^{(w)}} \log \pi_{\xi_k}(x_i)$$

based on the quantities involved in $\mathcal{Q}_n(\theta, \theta^{(w)})$ that depend on ξ_k. The update for $\xi_k^{(w)}$ therefore requires to solve the following convex minimization problem

$$\xi_k^{(w+1)} \in \operatorname{argmin}_{\xi_k \in \mathbb{R}^p} P^{(w)}_{n,k}(\xi_k) + \zeta_{1,k} \|\xi_k\|_{\text{en},\eta}. \tag{20}$$

It looks like the logistic regression objective, where labels are not fixed but softly encoded by the expectation step (computation of $\hat{\pi}_{ik}^{\theta^{(w)}}$ in (17)). We then choose to solve (20) using the L-BFGS-B algorithm [**?**] 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 penalization, which is not differentiable, we use the trick borrowed from **?**: 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 (20) as the following differentiable problem with box constraints

$$\begin{aligned} \text{minimize} \quad & P^{(w)}_{n,k}(\xi_k^+ - \xi_k^-) + \zeta_{1,k}\big((1-\eta)\sum_{j=1}^{p}(\xi^+_{k,j} + \xi^-_{k,j}) + \frac{\eta}{2}\|{\xi_k}^+ - {\xi_k}^-\|_2^2\big) \\ \text{subject to} \quad & \xi^+_{k,j} \geq 0 \text{ and } \xi^-_{k,j} \geq 0 \text{ for } j = 1, \ldots, p \end{aligned} \tag{21}$$

where $\xi_k^\pm = (\xi^\pm_{k,1}, \ldots, \xi^\pm_{k,p})^\top$. The L-BFGS-B solver requires the exact value of the gradient, which is easily given by

$$\frac{\partial P^{(w)}_{n,k}(\xi_k)}{\partial \xi_k} = -n^{-1} \sum_{i=1}^{n} \hat{\pi}_{ik}^{\theta^{(w)}} \big(1 - \pi_{\xi_k}(x_i)\big) x_i. \tag{22}$$

In practice, we use the `Python` solver `fmin_l_bfgs_b` from `scipy.optimize` [**?**]. Similarly to the strategy of increasing N with the iterations w for computational reasons, it is not clever to use a too small solver tolerance in the early steps of the algorithm, where we call tolerance the value for which iterations stop when the stopping criterion is below it. We then decrease the L-BFGS-B tolerance (which we denote *tol* in Algorithm 3) with every increase of N.

Now, for the update of $\beta_k^{(w)}$ for $k = 0, \ldots, K-1$, we similarly denote

$$
\begin{aligned}
R_{n,k}^{(w)}(\beta_k) = -n^{-1} \sum_{i=1}^{n} \hat{\pi}_{ik}^{\theta^{(w)}} \Big[& \delta_i \hat{\mathbb{E}}_{\theta^{(w)}}[\tilde{g}^2_{t_i, \gamma_k^{(w)}, \beta_k}(b_i) | t_i, \delta_i, y_i] \\
& - \sum_{j=1}^{J} \lambda_0^{(w)}(t_j) \hat{\mathbb{E}}_{\theta^{(w)}}[\tilde{g}^1_{i, t_j, \gamma_k^{0(w)}, \gamma_k^{(w)}, \beta_k}(b_i) | t_i, \delta_i, y_i] \mathbb{1}_{\{t_j \leq t_i\}} \\
& + (y_i \odot \Phi_i^{(w)})^\top \hat{\mathbb{E}}_{\theta^{(w)}}[M_{ik} | t_i, \delta_i, y_i] \\
& - \hat{\mathbb{E}}_{\theta^{(w)}}[c_{\phi^{(w)}}(M_{ik}) | t_i, \delta_i, y_i] \Big]
\end{aligned}
$$

based on the quantities involved in $\mathcal{Q}_n(\theta, \theta^{(w)})$ depending on β_k, with $\tilde{g}^1_{i, t_j, \gamma_k^{0(w)}, \gamma_k^{(w)}, \beta_k}$ and $\tilde{g}^2_{t_i, \gamma_k^{(w)}, \beta_k}$ defined in (18). The integration over the survival process has been replaced with a finite sum over the process evaluated at the distinct event times $(t_1, \ldots, t_J)$ with $J \in \mathbb{N}_+$, since $\lambda_0^{(w)}$ defined in (32) is always zero except at observed event times. The update for $\beta_k^{(w)}$ therefore requires to solve the following convex minimization problem

$$
\beta_k^{(w+1)} \in \operatorname{argmin}_{\beta_k \in \mathbb{R}^q} R_{n,k}^{(w)}(\beta_k) + \zeta_{2,k} \|\beta_k\|_{\text{sgl}_1, \tilde{\eta}}. \tag{23}
$$

The previous trick we used to handle the non-differentiability of the ℓ_1 part in the elastic net is no longer sufficient for (23), since the ℓ_2 norm in the sparse group lasso is not squared, leading to a non-differentiability at 0 (which is precisely the criterion exploited by the penalty to set some groups of coefficients to exactly 0 [**?**]). Then, we choose to solve problem (23) using the proximal gradient descent [**?**] described in Algorithm 2.

Algorithm 2: Proximal gradient descent $\text{ISTA}_{\text{sgl}_1}(v, f, \tilde{\eta}, \zeta)$

Input: A vector v, a function f (its gradient), hyper-parameters $\tilde{\eta}$, *tol* and ζ
Output: $\flat^{\breve{w}^{max}}$

Define $p_\varrho(y) = \operatorname{argmin}_u \left\{ \frac{1}{2\varrho} \left\| u - \left(y - \varrho \frac{\partial f}{\partial u}(y)\right) \right\|_2^2 + \zeta \|u\|_{\text{sgl}_1, \tilde{\eta}} \right\}$

Initialize $\flat^{(0)} = v$, $\varrho_0 > 0$, some $\epsilon > 1$

for $\breve{w} = 0, \ldots, \breve{w}^{max}$ **do**

  Find the smallest nonnegative integer $i^{\breve{w}}$ such that with $\tilde{\varrho}_{\breve{w}} = \epsilon^{i^{\breve{w}}} \varrho_{\breve{w}}$

  $$f(p_{\tilde{\varrho}_{\breve{w}}}(\flat^{(\breve{w})})) \leq f(\flat^{(\breve{w})}) + \left\langle \left(p_{\tilde{\varrho}_{\breve{w}}}(\flat^{(\breve{w})}) - \flat^{(\breve{w})}\right), \frac{\partial f}{\partial u}(\flat^{(\breve{w})}) \right\rangle + \frac{1}{2\tilde{\varrho}_{\breve{w}}} \|p_{\tilde{\varrho}_{\breve{w}}}(\flat^{(\breve{w})}) - \flat^{(\breve{w})}\|_2^2$$

  Set $\varrho_{(\breve{w}+1)} = \tilde{\epsilon}^{i^{\breve{w}}} \varrho_{\breve{w}}$

  $$\flat^{(\breve{w}+1)} \in \operatorname{argmin}_u \left\{ \frac{1}{2\varrho_{(\breve{w}+1)}} \left\| u - \left(\flat^{(\breve{w})} - \varrho_{(\breve{w}+1)} \frac{\partial f}{\partial u}(\flat^{(\breve{w})})\right) \right\|_2^2 + \zeta \|u\|_{\text{sgl}_1, \tilde{\eta}} \right\}$$

  if

  $$:= \text{prox}_{\text{sgl}_1, \tilde{\eta}, \zeta}\left(\flat^{(\breve{w})} - \varrho_{(\breve{w}+1)} \frac{\partial f}{\partial u}(\flat^{(\breve{w})})\right)$$

  $$\frac{\|\flat^{(\breve{w}+1)} - \flat^{(\breve{w})}\|_2}{\varrho_{(\breve{w}+1)}} < tol$$ **then**

    Convergence

Return: $\flat^{\breve{w}^{max}}$

TODO w/ VT: note on $\breve{w}^{max}$ and $\varrho_{\breve{w}}$. Lemma 1 bellow states that one can easily compute the proximal operator of the sparse group lasso defined in Algorithm 2.

Lemma 1. *The proximal operator of the sparse group lasso can be expressed as the composition of the group lasso and the lasso proximal operators, that is*

$$\text{prox}_{\text{sgl}_1,\tilde{\eta},\zeta} = \text{prox}_{\zeta\tilde{\eta}\sum_{l=1}^{L}\|\cdot\|_{2,l}} \circ \text{prox}_{\zeta(1-\tilde{\eta})\|\cdot\|_1}.$$

The two proximal operators on the right hand side of Lemma 1 are both well known analytically [**?**] and tractable. We used the `Python` library `copt` for the implementation of proximal gradient descent [**?**] and we propose in our package an implementation for the proximal operator of the sparse group lasso based on Lemma 1. A proof of Lemma 1 is proposed in Appendix A. Hence, we update $\beta_k^{(w)}$ with the following step

$$\beta_k^{(w+1)} = \text{ISTA}_{\text{sgl}_1}(\beta_k^{(w)}, R_{n,k}^{(w)}, \tilde{\eta}, \zeta_{2,k}). \tag{24}$$

Note that the warmstart initialization is left to the user's choice, as well as the ISTA acceleration [**?**], since warmstart can diminish acceleration effectiveness [**?**]. The gradient of $R_{n,k}^{(w)}$ for the Gaussian case is here given by

$$\begin{aligned}\frac{\partial R_{n,k}^{(w)}(\beta_k)}{\partial \beta_k^l} = -n^{-1}\sum_{i=1}^{n}\hat{\pi}_{ik}^{\theta^{(w)}}\Big[\sum_{a=1}^{\mathcal{A}}\gamma_{k,a}^{l\ (w)}\Big(\delta_i\hat{\mathbb{E}}_{\theta^{(w)}}[\frac{\partial\varphi_a(t_i,\beta_k^l,b_i^l)}{\partial\beta_k^l}|t_i,\delta_i,y_i] \\ - \sum_{j=1}^{J}\lambda_0^{(w)}(t_j)\hat{\mathbb{E}}_{\theta^{(w)}}[\frac{\partial\varphi_a(t_j,\beta_k^l,b_i^l)}{\partial\beta_k^l}\tilde{g}^1_{i,t_j,\gamma_k^{0(w)},\gamma_k^{(w)},\beta_k}(b_i)|t_i,\delta_i,y_i]\mathbb{1}_{\{t_j\le t_i\}}\Big) \\ + U_{il}^\top {\phi_l^{(w)}}^{-1} I_{n_{il}}(y_i^l - U_{il}\beta_k^l - V_{il}\hat{\mathbb{E}}_{\theta^{(w)}}[b_i|t_i,\delta_i,y_i])\Big]\end{aligned} \tag{25}$$

for all $l = 1,\dots,L$. Note that the gradients of the considered φ_a functions with respect to β_k^l are given in Table 1.

Concerning the update of $\gamma_k^{0(w)}$ and $\gamma_k^{(w)}$ for $k = 0,\dots,K-1$, let us follow the same strategy by denoting

$$\begin{aligned}Q_{n,k}^{(w)}(\gamma_k^0,\gamma_k) = -n^{-1}\sum_{i=1}^{n}\hat{\pi}_{ik}^{\theta^{(w)}}\Big[\delta_i\hat{\mathbb{E}}_{\theta^{(w)}}[\log\circ\tilde{g}^1_{i,t_i,\gamma_k^0,\gamma_k,\beta_k^{(w+1)}}(b_i)|t_i,\delta_i,y_i] \\ - \sum_{j=1}^{J}\lambda_0^{(w)}(t_j)\hat{\mathbb{E}}_{\theta^{(w)}}[\tilde{g}^1_{i,t_j,\gamma_k^0,\gamma_k,\beta_k^{(w+1)}}(b_i)|t_i,\delta_i,y_i]\mathbb{1}_{\{t_j\le t_i\}}\Big]\end{aligned}$$

based on the quantities involved in $\mathcal{Q}_n(\theta,\theta^{(w)})$ that depend on γ_k^0 and γ_k, with $\tilde{g}^1_{i,t_j,\gamma_k^0,\gamma_k,\beta_k^{(w+1)}}$ defined in (18). The update for $\gamma_k^{0(w)}$ therefore requires to solve the following minimization problem

$$\gamma_k^{0(w+1)} \in \text{argmin}_{\gamma_k^0\in\mathbb{R}^p} Q_{n,k}^{(w)}(\gamma_k^0,\gamma_k^{(w)}) + \zeta_{1,k}\|\gamma_k^0\|_{\text{en},\eta}. \tag{26}$$

We then rewrite problem (26) as the following differentiable problem with box constraints

$$\begin{aligned}&\text{minimize} && Q_{n,k}^{(w)}({\gamma_k^0}^+ - {\gamma_k^0}^-,\gamma_k^{(w)}) + \zeta_{1,k}\big((1-\eta)\sum_{j=1}^{p}({\gamma_{k,j}^0}^+ + {\gamma_{k,j}^0}^-) + \frac{\eta}{2}\|{\gamma_k^0}^+ - {\gamma_k^0}^-\|_2^2\big) \\ &\text{subject to} && {\gamma_{k,j}^0}^+ \ge 0 \text{ and } {\gamma_{k,j}^0}^- \ge 0 \text{ for } j = 1,\dots,p.\end{aligned} \tag{27}$$

The gradient is here given by

$$\frac{\partial Q_{n,k}^{(w)}(\gamma_k^0, \gamma_k{}^{(w)})}{\partial \gamma_k^0} = -n^{-1} \sum_{i=1}^{n} \hat{\pi}_{ik}^{\theta^{(w)}} \Big[\delta_i - \sum_{j=1}^{J} \lambda_0^{(w)}(t_j) \times \hat{\mathbb{E}}_{\theta^{(w)}}[\tilde{g}^1_{i,t_j,\gamma_k^0,\gamma_k{}^{(w)},\beta_k^{(w+1)}}(b_i)|t_i,\delta_i,y_i]\mathbb{1}_{\{t_j \le t_i\}} \Big] x_i. \tag{28}$$

Then, the update for $\gamma_k^{(w)}$ requires to solve the following minimization problem

$$\gamma_k^{(w+1)} \in \text{argmin}_{\gamma_k \in \mathbb{R}^{L\mathcal{A}}} \, Q_{n,k}^{(w)}(\gamma_k^{0(w+1)}, \gamma_k) + \zeta_{3,k} \|\gamma_k\|_{\text{sgl}_1,\tilde{\eta}}. \tag{29}$$

We treat problem (29) in the same way as (23), that is

$$\gamma_k^{(w+1)} = \text{ISTA}_{\text{sgl}_1}\big(\gamma_k^{(w)}, Q_{n,k}^{(w)}(\gamma_k^{0(w+1)}, \cdot), \tilde{\eta}, \zeta_{3,k}\big) \tag{30}$$

with the gradient being known since

$$\frac{\partial Q_{n,k}^{(w)}(\gamma_k^{0(w+1)}, \gamma_k)}{\partial \gamma_{k,a}^l} = -n^{-1} \sum_{i=1}^{n} \hat{\pi}_{ik}^{\theta^{(w)}} \Big[\delta_i \hat{\mathbb{E}}_{\theta^{(w)}}[\varphi_a(t_i, \beta_k^{l\,(w+1)}, b_i^l)|t_i,\delta_i,y_i] - \sum_{j=1}^{J} \lambda_0^{(w)}(t_j) \hat{\mathbb{E}}_{\theta^{(w)}}[\varphi_a(t_j, \beta_k^{l\,(w+1)}, b_i^l) \tilde{g}^1_{i,t_j,\gamma_k^{0(w+1)},\gamma_k,\beta_k^{(w+1)}}(b_i)|t_i,\delta_i,y_i]\mathbb{1}_{\{t_j \le t_i\}} \Big] \tag{31}$$

for all $a = 1, \ldots, \mathcal{A}$ and $l = 1, \ldots, L$.

For the update of $\lambda_0^{(w)}$, we treat the jump size at the observed event times as parameters to be estimated [**?**]. The closed-form update is then given by

$$\lambda_0^{(w+1)}(t) = \frac{\sum_{i=1}^{n} \delta_i \mathbb{1}_{\{t=t_i\}}}{\sum_{i=1}^{n} \sum_{k=0}^{K-1} \hat{\pi}_{ik}^{\theta^{(w)}} \hat{\mathbb{E}}_{\theta^{(w)}} \Big[\tilde{g}^1_{i,t,\gamma_k^{(w+1)},\beta_k^{(w+1)}}(b_i)|t_i,\delta_i,y_i \Big] \mathbb{1}_{\{t_i \ge t\}}}, \tag{32}$$

which is a Breslow like estimator [**?**] adapted to our model.

Finally, for the update of $\phi^{(w)}$ and regarding Table 2, we focus on the Gaussian case being particularly useful in practice. It is obtained in closed-form by

$$\phi_l^{(w+1)} = \big(\sum_{i=1}^{n} n_i^l\big)^{-1} \sum_{i=1}^{n} \sum_{k=0}^{K-1} \hat{\pi}_{ik}^{\theta^{(w)}} \Big[\big(y_i^l - U_{il}\beta_k^{l\,(w+1)}\big)^\top \Big(y_i^l - U_{il}\beta_k^{l\,(w+1)} - 2V_{il}\hat{\mathbb{E}}_{\theta^{(w)}}[b_i^l|t_i,\delta_i,y_i] \Big) + \text{Tr}\left(V_{il}^\top V_{il} \hat{\mathbb{E}}_{\theta^{(w)}}[b_i^l b_i^{l\top}|t_i,\delta_i,y_i]\right) \Big]. \tag{33}$$

Appendix **??** gives some details about the way we implement the equations and their computation in practice.

Initialization. Our algorithm gives a local minimum, and it is clever to choose an initial value $\theta^{(0)}$ close to the final solution $\hat{\theta}$, so that the number of iterations required to reach convergence is reduced. We then give some details about the starting point $\theta^{(0)}$ of Algorithm 3. For all $k = 0, \ldots, K-1$, we first choose $\xi_k^{(0)} = \mathbf{0}_d$ and $\gamma_{k,a}^{l\,(0)} = 0$ for all $l = 1, \ldots, L$ and $a = 1, \ldots, \mathcal{A}$. Then, we initialize $\lambda_0^{(0)}$ and $\gamma_k^{0(0)}$ like if there is

no subgroup ($\gamma_{0,0}^{(0)} = \cdots = \gamma_{K-1,0}^{(0)}$) with a standard Cox proportional hazards regression with time-independent features using the `Python` library `tick`. Finally, the longitudinal submodels parameters $\beta_k^{(0)}$, $D^{(0)}$ and $\phi^{(0)}$ are initialized – again like if there is no subgroup ($\beta_0^{(0)} = \cdots = \beta_{K-1}^{(0)}$) – using a multivariate linear mixed model with an explicit EM algorithm (the details are given in Appendix D), being itself initialized with univariates fits using the `Python` package `statsmodels`.

The QNMCEM algorithm. Algorithm 3 describes the main steps of the resulting QNMCEM algorithm to solve the optimization problem (10). The *Condition* in algorithm 3 is $\text{cv}(\Delta_{\text{rel}}^{(w+1)}) > \text{cv}(\Delta_{\text{rel}}^{(w)})$, where $\text{cv}(\Delta_{\text{rel}}^{(w)})$ is coefficient of variation at the $(w+1)$-th iteration computed as

$$\text{cv}(\Delta_{\text{rel}}^{(w+1)}) = \frac{\text{sd}(\Delta_{\text{rel}}^{(w-1)}, \Delta_{\text{rel}}^{(w)}, \Delta_{\text{rel}}^{(w+1)})}{\text{mean}(\Delta_{\text{rel}}^{(w-1)}, \Delta_{\text{rel}}^{(w)}, \Delta_{\text{rel}}^{(w+1)})},$$

where sd(.) and mean(.) are the sample standard deviation and mean functions respectively and relative difference $\Delta_{\text{rel}}^{(w+1)}$ is given by

$$\Delta_{\text{rel}}^{(w+1)} = \max\left\{\frac{|\theta^{(w+1)} - \theta^{(w)}|}{|\theta^{(w)}| + \epsilon}\right\}$$

Algorithm 3: QNMCEM algorithm for the lights model inference

Input: Training data $\mathcal{D}_n$; initialize $N = 50L$; tuning hyper-parameters $(\zeta_{1,k}, \zeta_{2,k}, \zeta_{3,k})_{k=0,\ldots,K-1}$

Output: Last parameters $\hat{\theta} \in \mathbb{R}^{\vartheta}$

/* Initialization */

Compute the starting parameters $\theta^{(0)} \in \mathbb{R}^{\vartheta}$

for $w = 1, \ldots,$ *until convergence* **do**

  /* Monte Carlo E-step */

  Compute $\hat{\Lambda}_{ik,\tilde{g}_i}^{\theta^{(w)}}$ using (16) for the required functions $\tilde{g}_i$

  Compute $\hat{\pi}_{ik}^{\theta^{(w)}}$ using (17)

  if *Condition* **then**

    $N \leftarrow N+$

    $tol \leftarrow tol-$

  /* Quasi-Newton M-step */

  Update $D^{(w+1)}$ using (19)

  Update $(\xi_k^{(w+1)})_{k=0,\ldots,K-1}$ by solving (21)

  Update $(\beta_k^{(w+1)})_{k=0,\ldots,K-1}$ by solving (24)

  Update $(\gamma_k^{0(w+1)})_{k=0,\ldots,K-1}$ by solving (27)

  Update $(\gamma_k^{(w+1)})_{k=0,\ldots,K-1}$ by solving (30)

  Update $\lambda_0^{(w+1)}$ using (32)

  Update $\phi^{(w+1)}$ using (33)

Return: $\hat{\theta}$

3.3 The case $K = 2$

In many practical applications – including those addressed in Section 6 – we are interested in identifying one subgroup of the population with a high risk of adverse event compared to the others. Then, in the following, we consider $G \in \{0, 1\}$ where $G = 1$ means high-risk of early adverse event and $G = 0$ means low risk. To simplify notations, let us set $\xi = \xi_1$, $\pi_\xi(x)$ the conditional probability that a subject belongs to the group with high risk of adverse event given its time-independent features x, and $\zeta_1 = \zeta_{1,1}$. In this context, which is the one we consider in practice in Sections 5 and 6, we suppose that the "right" penalty strength for ξ_k and γ_k does not depend on k, so that we denote $\zeta_2 = \zeta_{2,0} = \zeta_{2,1}$ and $\zeta_3 = \zeta_{3,0} = \zeta_{3,1}$.

Cross-validation procedure. The hyper-parameters $(\zeta_1, \zeta_2, \zeta_3)^\top \in \mathbb{R}_+^3$ are then tuned during a 10-fold randomized search cross-validation procedure [**?**] using predictive marker defined in Section 4.1 and the C-index score defined in Section 4.2. Random search is indeed more efficient in terms of computing times than classical grid search for hyper-parameters optimization and finds better models by effectively searching in a larger configuration space, which is appropriate to our high-dimensional problem. Moreover, the search interval for hyper-parameter ζ_1 is automatically computed in a data driven way described in Appendix B, while the search intervals for ζ_2 and ζ_3 are determined empirically.

4 Performance evaluation

In this section, we first present the proposed evaluation strategy to assess real-time prognostic prediction performances. We then introduce the appropriate metric used to this end: the C-index ; and finally, the considered models for performance comparisons.

4.1 Evaluation strategy: the real-time prediction paradigm

When validating predictions or comparing performances of competing models in a survival context, one can be interested either in the discriminative power of the predictive rule and use concordance measures such as the C-index [**?**] (defined in Section 4.2), or in the predictive accuracy of the rule [**?**]. Developments in the field of joint modeling have primarily focused on modeling and estimation, and most studies do not consider goodness-of-fit nor generalization power in a prognostic prediction perspective [**?**]. However, with the prospect of using prognostic prediction in real-time on a daily basis, practitioners will naturally require predictive prognostic tools to evaluate models fit and compare models in terms of discriminative power. So we choose to put ourselves in this paradigm.

Predictive marker. The question now is to choose a discriminative marker rule for risk prediction to be used in the cross-validation procedure for selecting the best regularization hyper-parameters, and of course to be used as final risk prediction after hyper-parameters fine-tuning. The lights model has the ability to produce prognostic predictions that can be dynamically updated according to the observed trajectory of the features. Once the model is trained (so that one obtains $\hat{\theta}$ from (10) using our QNMCEM algorithm introduced in Section 3.2), posterior risk classification for subject i and group k can be made through $\hat{\pi}_{ik}^{\hat{\theta}}$ (see (17)). Similar posterior probabilities have been considered for goodness-of-fit evaluation in other JLCMs models, see **?** for instance.

But in the real-time prediction paradigm, it makes no sense from a practical point of view to use this marker rule as is for risk prediction, since the latter requires to know the

survival labels (t_i, δ_i), being intrinsically unknow in a context where we want to perform real-time risk prediction. Denoting t_i^{max} the time for subject i when one wants to perform the risk prediction – so in practice, the time up to which one has data measurements for Y_i, say the "present" time for prediction – a natural and appropriate marker rule of the lights model for subject i being in subgroup k would rather be

$$\hat{\mathcal{R}}_{ik} = \frac{\pi_{\hat{\xi}_k}(x_i)\hat{f}(t_i^{max}, y_i | G_i = k; \hat{\theta})}{\sum_{k=0}^{K-1} \pi_{\hat{\xi}_k}(x_i)\hat{f}(t_i^{max}, y_i | G_i = k; \hat{\theta})}, \tag{34}$$

with

$$\hat{f}(t_i^{max}, y_i | G_i = k; \hat{\theta}) = \frac{1}{2N} \sum_{\breve{b} \in \hat{S}} f(t_i^{max}, 0, y_i | \breve{b}, G_i = k; \hat{\theta}),$$

where $\hat{S}$ is obtained with Algorithm 1 using $\hat{D}$ (that is $S^{(w_{max})}$ for the last iteration w_{max}), and the density f in the rigth hand side of the previous equation being defined in (7). Note that marker (34) is in fact obtained from the definition of $\hat{\pi}_{ik}^{\hat{\theta}}$ in (17) taking $(t_i, \delta_i) = (t_i^{max}, 0)$, and turns out to be an estimate of the following posterior probability

$$\mathbb{P}_\theta[G_i = k | T_i^\star > t_i^{max}, y_i, x_i].$$

4.2 The C-index metric

We detail in this section the metric considered to evaluate risk prediction performances. Let us recall that in practice, one considers the case $K = 2$, see Section 3.3. In this context, the predictive marker for the lights model is

$$\hat{\mathcal{R}}_i^{\text{lights}} = \frac{\pi_{\hat{\xi}}(x_i)\hat{f}(t_i^{max}, y_i | G_i = 1; \hat{\theta})}{\pi_{\hat{\xi}}(x_i)\hat{f}(t_i^{max}, y_i | G_i = 1; \hat{\theta}) + \big(1 - \pi_{\hat{\xi}}(x_i)\big)\hat{f}(t_i^{max}, y_i | G_i = 0; \hat{\theta})}.$$

Then, let us denote by $\hat{\mathcal{R}}$ the marker under study in the general case, and assume that it is measured once at $t = t^{max}$. A common concordance measure that does not depend on time is the C-index [**?**] defined by

$$\mathcal{C} = \mathbb{P}[\hat{\mathcal{R}}_i > \hat{\mathcal{R}}_j | T_i^\star < T_j^\star],$$

with $i \neq j$ two independent subjects (which 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 $\mathcal{C}_\tau$ defined by

$$\mathcal{C}_\tau = \mathbb{P}[\hat{\mathcal{R}}_i > \hat{\mathcal{R}}_j | T_i < T_j, T_i < \tau],$$

with τ corresponding to the fixed and prespecified follow-up period duration [**?**]. A Kaplan-Meier estimator for the censoring distribution leads to a nonparametric and consistent estimator of $\mathcal{C}_\tau$ [**?**], already implemented in the `Python` package `lifelines`. Hence in the following, we consider the C-index metric to assess performances.

Note that other metrics have been proposed to compare individual risk predictions. In **?** for instance, authors use dynamic prediction accuracy curves (of AUC or Brier score) with the idea of moving what is called the "landmark time", that is the time at which predictions are made and on which the amount of available information depends, corresponding to our t^{max}. But comparing curves in an automatic way (in a cross-validation process for instance) is never easy, plus the comparisons proposed in the aforementioned paper require

a priori choices of time window thresholds (in particular to get back to binary predictions, being always questionable in survival analysis [**?**]), which is not the case of the C-index metric. Nevertheless, the idea of chosing t^{max} when it comes to simulate data for models performances comparisons will be discussed in Section 5, and it is an interesting notion to consider since it can be sensed that the closer t^{max} gets to $T^\star$, the easier it will be for a model to make good predictions.

4.3 Competing models

In this section, we briefly introduce the models we consider for performance comparisons in the simulation study as well as in the applications on real datasets in Section 6. We also give the corresponding marker rule for each model, using our notations.

Landmark Cox model. The first model we consider as a baseline is the well known Cox PH model with time-independent features, also known in the joint modeling context as the "landmark" model, in which we include basic time-independent features extracted from longitudinal processes and computed at the prediction time [REF]. In the Cox PH model introduced in **?**, a parameter vector β is estimated by minimizing the partial log-likelihood given by

$$\ell_n^{\text{cox}}(\beta) = n^{-1} \sum_{i=1}^{n} \delta_i \big(x_i^\top \beta - \log \sum_{i' : t_{i'} \geq t_i} \exp(x_{i'}^\top \beta) \big).$$

For each time-dependant feature, we include subject-specific random effects, values of longitudinal processes at baseline and at time t^{max}, slope of longitudinal processes at time t^{max} and area under of longitudinal processes from baseline up to t^{max}. The predictive marker for subject i is the regular risk score [**?**], that is $\hat{\mathcal{R}}_i^{\text{cox}} = \exp(x_i^\top \hat{\beta}^{\text{cox}})$, with

$$\hat{\beta}^{\text{cox}} \in \text{argmin}_{\beta \in \mathbb{R}^\vartheta} -\ell_n^{\text{cox}}(\beta).$$

We use the `Python` package `tick` for the estimation of the previous quantity. Ties are handled via the Breslow approximation of the partial likelihood [**?**].

Penalized Cox model. We then consider a penalized version of the Cox model, including now much more time-independent features automatically extracted from longitudinal processes using the `Python` package `tsfresh` [**?**]. The predictive marker for the a subject i is $\hat{\mathcal{R}}_i^{\text{cox EN}} = \exp(x_i^\top \hat{\beta}^{\text{cox EN}})$, with

$$\hat{\beta}^{\text{cox EN}} \in \text{argmin}_{\beta \in \mathbb{R}^\vartheta} -\ell_n^{\text{cox}}(\beta) + \xi \big((1-\eta) \|\beta\|_1 + \frac{\eta}{2} \|\beta\|_2^2 \big)$$

the estimate obtained using elastic net penalization (still with the `tick` package), and where ξ is chosen by the a 10-fold cross-validation procedure, for a given $\eta \in [0, 1]$.

The time-dependent Cox model. A classical extension of the Cox model supposes that features depend on time [**?**] (and then loosing the risk proportionality property), that is with our notations

$$\lambda_i(t) = \lambda_0(t) \exp \big(x_i^\top \gamma_0 + y_i(t_i^{max})^\top \beta \big).$$

We use the `survival R` package [**?**], and we consider the natural predictive marker for a subject i, that is the corresponding risk score given by

$$\hat{\mathcal{R}}_i^{\text{cox t-d}} = \exp \big(x_i^\top \hat{\gamma}_0 + y_i(t)^\top \hat{\beta} \big).$$

Multivariate joint latent class model. We consider a multivariate version of JLCMs implemented in the `R` package `lcmm` [**?**]. Contrary to the lights model, there is no shared associations between the longitudinal models and the survival model. Given the group membership, each submodel is assumed to be independent. In this context, the predictive marker for subject i is

$$\hat{\mathcal{R}}_i^{\text{lcmm}} = \frac{\pi_{\hat{\xi}}(x_i) f(t_i^{max}, y_i | G_i = 1; \hat{\theta})}{\pi_{\hat{\xi}}(x_i) f(t_i^{max}, y_i | G_i = 1; \hat{\theta}) + \big(1 - \pi_{\hat{\xi}}(x_i)\big) f(t_i^{max}, y_i | G_i = 0; \hat{\theta})}.$$

Multivariate shared random effect model. In this context, we consider the `JMbayes` package [**?**]. The predictive marker associated with this model for subject i is naturally given by the corresponding risk score

$$\hat{\mathcal{R}}_i^{\text{JMbayes}} = \exp\Big\{ x_i^\top \hat{\gamma}_0 + \sum_{l=1}^{L} \sum_{a=1}^{\mathcal{A}} \widehat{\gamma_{k,a}^l}^\top \varphi_a(t_i^{max}, \hat{\beta}^l, b_i^l) \Big\}.$$

`@Antoine : est-ce qu'il peut prendre des time-indep feat ? et quelles sont les assos ?`

5 High-dimensional simulation study

In this section, we provide details regarding data generation, followed by the results of the extensive Monte Carlo simulation study to examine our method and compare it with state-of-the-art.

5.1 Simulation design

In order to assess the proposed method, we run an extensive Monte Carlo simulation study in the case $K = 2$ (see Section 3.3). We first choose a coefficient vector

$$\xi = (\underbrace{\varsigma_1, \dots, \varsigma_1}_{s}, 0, \dots, 0) \in \mathbb{R}^p, \tag{35}$$

with $\varsigma_1 \in \mathbb{R}$ being the value of the active coefficients and $s \in \{1, \dots, p\}$ a sparsity parameter chosen such that a proportion r_s of time-independent features are actually active, that is $s = \lfloor p \times r_s \rfloor$. For a desired high-risk subjects proportion $\pi_1 \in [0, 1]$, the high-risk subjects index set is given by

$$\mathcal{H} = \big\{ \lfloor \pi_1 n \rfloor \text{ random samples without replacement} \big\} \subset \{1, \dots, n\}.$$

For the generation of the time-independent features matrix, we take

$$[x_{ij}] \in \mathbb{R}^{n \times p} \sim \mathcal{N}\big(0, \Sigma_1(\rho_1)\big),$$

with $\Sigma_1(\rho_1)$ a $(p \times p)$ Toeplitz covariance matrix [**?**] with correlation $\rho_1 \in (0, 1)$, that is $\Sigma_1(\rho_1)_{jj'} = \rho_1^{|j - j'|}$. We then add a $gap \in \mathbb{R}^+$ value for subjects $i \in \mathcal{H}$ and subtract it for subjects $i \notin \mathcal{H}$, only on active features, that is

$$x_{ij} \leftarrow x_{ij} + (-1)^{\mathbb{1}_{\{i \notin \mathcal{H}\}}} gap \text{ for } j = 1, \dots, s.$$

Note that this is equivalent to generate the time-independent features according to a gaussian mixture. We finally normalize each time-independent features for practical interpretations. Then, we generate $G_i \sim \mathcal{B}\big(\pi_\xi(x_i)\big)$, where $\pi_\xi(x_i)$ is computed given Equation (2) and $\mathcal{B}(\alpha)$ denotes the Bernoulli distribution with parameter $\alpha \in [0,1]$.

Now, concerning the simulation of the longitudinal data, the idea is to sample from multivariate normal distributions. We use linear time-varying features such that

$$Y_i^l(t) = \sum_{k=0}^{K-1} \mathbb{1}_{\{G_i=k\}} \big((1,t)^\top \beta_k^l + (1,t)^\top b_i^l + \epsilon_i^l(t)\big)$$

where $t \geq 0$, the error term $\epsilon_i^l(t) \sim \mathcal{N}(0, \sigma_l^2)$, the global variance-covariance matrix for the random effects components (4) 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

$$\beta_k^l \sim \mathcal{N}\Big(\mu_k, \begin{bmatrix} \rho_3 & 0 \\ 0 & \rho_3 \end{bmatrix}\Big)$$

for $k \in \{0,1\}$ and with correlation $\rho_3 \in (0,1)$, so that β_k^l coefficients are both positive for high-risk subjects ($G_i = 1$), which favours increasing trajectories with positive values, and negative otherwise ($G_i = 0$), which favours decreasing trajectories with negative values. This choice makes sens regarding that for simplicity, we take joint association parameters being all positive, see (36).

Most joint modeling simulation studies use fixed time points [**?**], that is $n_i^l = n_i^{l'}$ and $t_{ij}^l = t_{ij}^{l'}$ for all $(l,l') \in \{1,\dots,L\}^2$ and $j = 1,\dots,n_i^l$ for a given subject i, but it does not reflect actual behavior in practice, since data measurements of the different longitudinal processes are barely done simultaneously. For this reason, the simulation of realistic longitudinal features is not trivial and we use Hawkes processes [**?**] with exponential kernels to generate measurement times, which is a family of counting process with an autoregressive intensity. Hawkes processes model self-exciting behavior, that is in our case, when the measurement of one longitudinal process makes future measurements of the process – as well as those of others processes, in a multivariate way – more likely to happen. Namely, for a subject i, times $\{t_{ij}^l\}_{j\geq 1}$ for processes $l = 1,\dots,L$ are simulated using a multivariate Hawkes process $N_{it} = [N_{it}^1 \cdots N_{it}^L]$ with $t \geq 0$ and $N_{it}^l = \sum_{j\geq 1} \mathbb{1}_{\{t_{ij}^l \leq t\}}$. The process N_{it} is a multivariate counting process, whose components N_{it}^l have intensities

$$\lambda_i^l(t) = \Upsilon_l + \sum_{l'=1}^{L} \sum_{j\geq 1} A_{ll'} \upsilon \exp\big(-\upsilon(t - t_{ij}^{l'})\big)$$

for $l = 1,\dots,L$. $\Upsilon_l \geq 0$ is called baseline intensity, and corresponds to the exogenous probability of having a measurement for process l. We sample $\Upsilon_l \sim \mathcal{U}\big([0.1,1]\big)$] which produces unbalanced baselines, where $\mathcal{U}\big([a,b]\big)$ stands for the uniform distribution on a segment $[a,b]$. The matrix $A = [A_{ll'}]_{1\leq l,l'\leq L}$ is the adjacency matrix such that $A_{ll'} \geq 0$ quantifies the impact of past measurement time of process l' on the measurement time of process l, and $\upsilon \geq 0$ is a memory parameter. We sample $A_{ll'} \sim \mathcal{U}\big([0.1,0.2]\big)$, but then enforce A to be sparse (with a chosen density of 0.3) by randomly zeroing non-diagonal coefficients. Simulation of the Hawkes processes is achieved using the `tick` library [**?**]. An example of simulated longitudinal features for an individual is displayed in Figure 1(b) below.

To generate survival times, we choose a risk model of the form

$$\lambda_i(t|G_i = k) = \lambda_0(t) \exp\Big\{x_i^\top \xi + \sum_{l=1}^{L} \gamma_{k,1}^l(\beta_{k,1}^l + \beta_{k,2}^l t + b_{i,1}^l + b_{i,2}^l t) + (\gamma_{k,2,1}^l b_{i,1}^l + \gamma_{k,2,2}^l b_{i,2}^l) + \gamma_{k,3}^l(\beta_{k,2}^l + b_{i,2}^l)\Big\},$$

which corresponds to (5) taking $\gamma_k^0 = \xi$ and the first three functionals described in Table 1 as shared associations. Now, for the choice of the association parameters, we want to induce sparsity (similarly to (35)), so we set

$$\mathcal{S}_k = \Big\{k\lfloor\frac{Lr_s}{K}\rfloor + 1, \dots, (k+1)\lfloor\frac{Lr_s}{K}\rfloor\Big\}$$

and

$$\gamma_{k,a}^l = \varsigma_2\big(\mathbb{1}_{\{l \leq (k+1)\lfloor Lr_s/K\rfloor\}} + \mathbb{1}_{\{l \in \mathcal{S}_k\}}\big)\mathbf{1}_{\iota_a}, \tag{36}$$

so that there is a proportion r_s of active longitudinal features among the L, and the longitudinal effects are distinct for each subgroup $k = 0, \dots, K-1$. Note that one can write

$$\lambda_i(t|G_i = k) = \lambda_0(t) \exp\big\{\iota_{i,k,1} + \iota_{i,k,2} t\big\},$$

being a Cox model with a linear time-varying feature. We choose a Gompertz distribution [**?**] for the baseline, that is

$$\lambda_0(t) = \kappa_1 \kappa_2 \exp(\kappa_2 t) \tag{37}$$

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 [**?**] with a rich history in describing mortality curves. One can now generate survival times explicitly as

$$T_i^\star | G_i = k \sim \frac{1}{\iota_{i,k,2} + \kappa_2} \log\Big(1 - \frac{(\iota_{i,k,2} + \kappa_2)\log U_i}{\kappa_1 \kappa_2 \exp \iota_{i,k,1}}\Big) \tag{38}$$

where $U_i \sim \mathcal{U}\big([0,1]\big)$ (see Appendix C for the derivation of this expression). 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 $r_c \in [0,1]$. Finally, the prediction time for subject i is draw according to

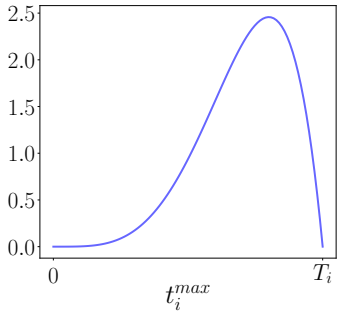
$$t_i^{max} \sim T_i \times \big(1 - \text{Beta}(\nu_1, \nu_2)\big) \tag{39}$$

with $(\nu_1, \nu_2) \in \mathbb{R}_+^2$ two shape parameters. Figure 1(a) shows the corresponding probability density function for t_i^{max} with some explanations. We then truncate all Y_i measurements being after t_i^{max}, since it represents the "present" time, so the time up to which one has longitudinal data. The choice of all hyper-parameters is driven by the applications on real data presented in Section 6, and summarized in Table 3. Figure 1(b) gives an example of data generated according to the design we have just described.

Following the real-time prediction paradigm presented in Section 4.1, we compare the predictive performances of the lights model with the competing ones introduced in Section 4.3 in terms of C-index (see Section 4.2). We also want to assess the stability of the lights model in terms of feature selection, but we do not compare this aspect with state-of-the-art since other models are not designed to perform feature selection, and comparisons would be unfair. To this end, we follow the same simulation procedure

Table 3: Hyper-parameter choices for simulation. Let us also precise that we take $n \in [200, 4000]$ and $(p, L) \in [3, 300]^2$.

π_1	(ρ_1, ρ_2, ρ_3)	μ_0	μ_1	gap	σ_l^2	υ	(ν_1, ν_2)	(κ_1, κ_2)	$(\varsigma_1, \varsigma_2)$	r_c	r_s	$\eta = \tilde{\eta}$
0.4	$(, 10^{-3},)$	$\begin{pmatrix} 1 \\ 0.3 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0.5 \end{pmatrix}$	0.5	4	3	(2, 5)	$(10^{-3}, 0.1)$	(1, 0.1)	0.3	0.4	0.1



(a) Density of the law used to simulate t_i^{max} according to (39) with $(\nu_1, \nu_2) = (2, 5)$, to mimic the fact that in practice, one has access to a reasonable amount of longitudinal data before making a prediction.

(b) ex- em- ple simu

Figure 1: Illustrations of simulated data.

explained in the previous lines and for each simulation case, we use the following approach to evaluate the variable selection power of the model. Denoting

$$\tilde{\xi}_j = \frac{|\hat{\xi}_j|}{\max_{j=1,\dots,p} |\hat{\xi}_j|} \qquad \text{and} \qquad \tilde{\gamma}_{k,a}^l = \frac{|\hat{\gamma}_{k,a}^l|}{\max_{a=1,\dots,\mathcal{A}} |\hat{\gamma}_{k,a}^l|}$$

for $k = 0, \dots, K-1$ and $l = 1, \dots, L$, and considering that $\tilde{\xi}_j$ and $\tilde{\gamma}_{k,a}^l$ are the predicted probability that the true ξ_j equals ς_1 and the predicted probability that the true $\gamma_{k,a}^l$ equals ς_2 respectively, then we are in a binary prediction setting and we use the two resulting AUC scores of this problem to evaluate the selection power of the time-independent and the time-dependent features respectively.

5.2 Results of simulation

Let us present now the simulation results.

6 Applications

In this section, we apply our lights method on two publicly available datasets and compare its performance with state-of-the-art methods.

6.1 The PBCseq dataset

This dataset is a follow-up to the original dataset [**??**]from the Mayo Clinic trial in primary biliary cirrhosis (PBC) of the liver conducted between 1974 and 1984. A total of 424 PBC

patients, referred to Mayo Clinic during that ten-year interval, met eligibility criteria for the randomized placebo controlled trial of the drug D-penicillamine. The first 312 cases in the data set participated in the randomized trial and contain largely complete data. The additional 112 cases did not participate in the clinical trial, but consented to have basic measurements recorded and to be followed for survival. Six of those cases were lost to follow-up shortly after diagnosis, so the data here are on an additional 106 cases as well as the 312 randomized participants. The dataset contains only baseline measurements of the laboratory paramters and contains multiple laboratory results, but only on the first 312 patients.

6.2 The MIMIC III dataset

The MIMIC III (Medical Information Mart for Intensive Care III) database is a large, freely-available hospital dataset containing de-identified data from over 40,000 patients. This data comes from patients who were admitted in critical care units to Beth Israel Deaconess Medical Center in Boston, Massachusetts from 2001 to 2012 [**?**]. The dataset was populated with data that had been acquired during routine hospital care, so there was no associated burden on caregivers and no interference with their workflow.

7 Conclusion

In this paper, a generalized joint model for high-dimensional multivariate longitudinal data and censored durations (lights) has been introduced, and a new efficient estimation algorithm (QNMCEM) has been derived, that considers a penalization of the likelihood in order to perform feature selection and to prevent overfitting.

Software

All the methodology discussed in this paper is implemented in `Python`. The code is available from `https://github.com/Califrais/lights` in the form of annotated programs, together with a notebook tutorial. A technical documentation is also provided in Appendix E.

Acknowledgements

Conflict of Interest: None declared.

Appendices

A Proof of Lemma 1

B Automatic cross-validation grid for ξ

Concerning the cross-validation procedure for tuning

$$(\zeta_{1,0}, \zeta_{2,0}, \zeta_{3,0}, \ldots, \zeta_{1,K-1}, \zeta_{2,K-1}, \zeta_{3,K-1})^\top \in \mathbb{R}_+^{3K},$$

we use a randomized search with the C-index metric (see Section 4.1) where for all $k = 0, \dots, K-1$ and $j \in \{1,2,3\}$, each $\zeta_{j,k}$ finds its candidates in the interval

$$[\zeta_{j,k}^{\max} \times 10^{-4}, \zeta_{j,k}^{\max}] \subset \mathbb{R},$$

with $\zeta_{j,k}^{\max}$ the interval upper bound computed hereafter.

Considering the convex minimization problem (20) at a given step w, let us denote $\zeta_{1,k}^1 \leq \zeta_{1,k}^2 \leq \cdots \leq \zeta_{1,k}^{\max}$ the randomly chosen candidate values for $\zeta_{1,k}$, such that at $\zeta_{1,k}^{\max}$, all coefficients $\hat{\xi}_{k,j}$ for all $j = 1, \dots, p$ are exactly zero. The KKT conditions [**?**] claim that

$$\begin{cases} \dfrac{\partial P_{n,k}^{(w)}(\hat{\xi}_k)}{\partial \xi_{k,j}} = \zeta_{1,k}\big((\eta-1)\ \mathrm{sgn}(\hat{\xi}_{k,j}) - \eta\hat{\xi}_{k,j}\big) & \forall j \in \hat{\mathcal{A}}_k \\ \left|\dfrac{\partial P_{n,k}^{(w)}(\hat{\xi}_k)}{\partial \xi_{k,j}}\right| < \zeta_{1,k}(1-\eta) & \forall j \notin \hat{\mathcal{A}}_k \end{cases},$$

where $\hat{\mathcal{A}}_k = \big\{j = 1, \dots, p : \hat{\xi}_{k,j} \neq 0\big\}$ is the active set of the $\hat{\xi}_k$ estimator, and for all $x \in \mathbb{R} \setminus \{0\}$, $\mathrm{sgn}(x) = \mathbb{1}_{\{x>0\}} - \mathbb{1}_{\{x<0\}}$. Then, using (22), one obtains

$$\hat{\xi}_{k,j} = 0 \ \Rightarrow\ \left| n^{-1} \sum_{i=1}^{n} \hat{\pi}_{ik}^{\theta^{(w)}} \big(1 - \pi_{\xi_k}(x_i)\big) x_{ij} \right| < \zeta_{1,k}(1-\eta)$$

for all $j = 1, \dots, p$. Hence, we choose the following upper bound for the randomly chosen candidate interval during the cross-validation procedure

$$\zeta_{1,k}^{\max} = \frac{1}{n(1-\eta)} \max_{j=1,\dots,p} \sum_{i=1}^{n} |x_{ij}|.$$

Similar strategy to automatically compute $\zeta_{2,k}^{\max}$ or $\zeta_{3,k}^{\max}$ is not easy since without additionnal hypothesis, the gradients 25, 28 and 31 are not upper bounded by a quantity that depends only on the data and not on the current iteration parameters estimates. We then choose those interval upper bounds empirically.

C Event time simulation

Let us derive here the expression (38) used to generate event times. Indeed, we can not refer to **?** since the intensity (37) of the baseline is not correct in this paper. Note that other errors had already been pointed in [**?**], for instance the fact that there is no closed-form expression with a Weibull baseline. Suppose that

$$\lambda(t) = \lambda_0(t) \exp(a + bt)$$

with $(a, b) \in \mathbb{R}^2$ and λ_0 defined in (37), then the cumulative hazard function writes

$$H(t) = \kappa_1 \kappa_2 e^a \int_0^t \exp\{(\kappa_2 + b)s\} \mathrm{d}s = \frac{\kappa_1 \kappa_2 e^a}{\kappa_2 + b} \big(\exp\{(\kappa_2 + b)t\} - 1\big).$$

Then, on has

$$t = \frac{1}{\kappa_2 + b} \log\left(1 + \frac{H(t)(\kappa_2 + b)}{\kappa_1 \kappa_2 e^a}\right),$$

and one can generate event times according to

$$T \sim \frac{1}{\kappa_2 + b} \log\left(1 - \frac{(\kappa_2 + b) \log U}{\kappa_1 \kappa_2 e^a}\right)$$

with $U \sim \mathcal{U}\big([0,1]\big)$. □

D Multivariate linear mixed model

Let us derive here the explicit EM algorithm for the multivariate linear mixed model used to initialized the longitudinal parameters $\beta_k^{(0)}$, $D^{(0)}$ and $\phi^{(0)}$ in the QNMCEM algorithm in Section 3.2, acting as if there is no subgroup ($\beta_0^{(0)} = \cdots = \beta_{K-1}^{(0)}$). For the sake of simplicity, let us denote here

$$\theta = (\beta^\top, \mathrm{vech}(D), \phi^\top)^\top \in \mathbb{R}^\vartheta$$

the parameter vector to infer. The conditional distribution of $y_i|b_i$ then writes

$$f(y_i|b_i;\theta) = \exp\big\{(\Sigma_i y_i)^\top M_i - c_\phi(M_i) + d_\phi(y_i)\big\},$$

where Σ_i is the diagonal matrix whose diagonal is Φ_i. The complete log-likelihood in this context writes

$$\begin{aligned}\ell_n^{\mathrm{comp}}(\theta) &= \ell_n^{\mathrm{comp}}(\theta; \mathcal{D}_n, \boldsymbol{b}) \\ &= \sum_{i=1}^n (\Sigma_i y_i)^\top M_i - c_\phi(M_i) + d_\phi(y_i) - \frac{1}{2}\big(r\log 2\pi + \log|D| + b_i^\top D^{-1} b_i\big).\end{aligned}$$

E-step. Supposing that we are at step $w+1$ of the algorithm, with current iterate denoted $\theta^{(w)}$, we need to compute the expected negative log-likelihood of the complete data conditional on the observed data and the current estimate of the parameters, which is given by

$$\mathcal{Q}_n(\theta, \theta^{(w)}) = \mathbb{E}_{\theta^{(w)}}[\ell_n^{\mathrm{comp}}(\theta)|\mathcal{D}_n].$$

Here, computing this quantity reduces to computing $\mathbb{E}_{\theta^{(w)}}[b_i|y_i]$ and $\mathbb{E}_{\theta^{(w)}}[b_i b_i^T|y_i]$ for $i = 1, \ldots, n$. 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

$$\Omega_i^{(w)} = ({V_i}^\top \Sigma_i^{(w)} V_i + {D^{(w)}}^{-1})^{-1} \quad \text{and} \quad \mu_i^{(w)} = \Omega_i^{(w)} {V_i}^\top \Sigma_i^{(w)}(y_i - U_i\beta^{(w)}).$$

Then, on 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

$$\theta^{(w+1)} \in \mathrm{argmin}_{\theta\in\mathbb{R}^\vartheta}\, \mathcal{Q}_n(\theta, \theta^{(w)}).$$

The parameters updates are then naturally given in closed form by zeroing the gradient. One obtains

$$\beta^{(w+1)} = \Big(\sum_{i=1}^n U_i^\top U_i\Big)^{-1} \sum_{i=1}^n \big[U_i^\top y_i - U_i V_i \mathbb{E}_{\theta^{(w)}}[b_i|y_i])\big],$$

$$\phi_l^{(w+1)} = \big(\sum_{i=1}^{n} n_i^l\big)^{-1} \sum_{i=1}^{n} \Big[(y_i^l - U_{il}\beta_l^{(w+1)})^\top (y_i^l - U_{il}\beta_l^{(w+1)} - 2V_{il}\mathbb{E}_{\theta^{(w)}}[b_i^l|y_i^l])) + \mathrm{Tr}\left(V_{il}^\top V_{il}\mathbb{E}_{\theta^{(w)}}[b_i^l {b_i^l}^\top|y_i^l]\right)\Big]$$

and

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

E Technical documentation of the `lights` package

E.1 `MLMM class`

This class implements an EM algorithm for fitting a multivariate linear mixed model used to initialize the MCQNEM algorithm. Let us introduce the list $\Omega^{(w)} = [\Omega_1^{(w)}, \ldots, \Omega_n^{(w)}]$, the matrices $\mu = [\mu_1, \ldots, \mu_n] \in \mathbb{R}^{r\times n}$, $U^l = [{U_{1l}}^\top \cdots {U_{nl}}^\top]^\top \in \mathbb{R}^{n_l \times q_l}$, $U = [{U_1}^\top \cdots {U_n}^\top]^\top \in \mathbb{R}^{\mathcal{N}\times q}$,

$$V^l = \begin{bmatrix} V_{1l} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & V_{nl} \end{bmatrix}, \quad V = \begin{bmatrix} V_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & V_n \end{bmatrix} \quad \text{and} \quad \Omega^{l\,(w)} = \begin{bmatrix} {\Omega_1^l}^{(w)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & {\Omega_n^l}^{(w)} \end{bmatrix}$$

that belong respectively in $\mathbb{R}^{n_l\times nr_l}$, $\mathbb{R}^{\mathcal{N}\times nr}$ and $\mathbb{R}^{nr_l\times nr_l}$, as well as the vectors $\tilde{\mu}^{(w)} = ({\mu_1^{(w)}}^\top \cdots \mu_n^\top)^{(w)^\top} \in \mathbb{R}^{nr}$, $(\tilde{\mu}^l)^{(w)} = \big((\mu_1^l)^{(w)^\top} \cdots (\mu_n^l)^{(w)^\top}\big)^\top \in \mathbb{R}^{nr_l}$, $y^l = ({y_1^l}^\top \cdots {y_n^l}^\top)^\top \in \mathbb{R}^{n_l}$ with $n_l = \sum_{i=1}^n n_i^l$ 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

$$D^{(w+1)} = n^{-1}\big(\mathrm{sum}(\Omega^{(w)}) + \mu^{(w)}{\mu^{(w)}}^\top\big).$$

And finally for the ϕ update, one has

$$\phi_l^{(w+1)} = n_l^{-1}\big[(y^l - U^l\beta_l^{(w+1)})^\top\big(y^l - U^l\beta_l^{(w+1)} - 2V^l(\tilde{\mu}^l)^{(w)}\big) + \mathrm{Tr}\big\{{V^l}^\top V^l\big(\Omega^{l\,(w)} + (\tilde{\mu}^l)^{(w)}{(\tilde{\mu}^l)^{(w)}}^\top\big)\big\}\big].$$