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

Simon Bussy*[1,2], Van Tuan Nguyen[2,3], Antoine Barbieri[4], Sarah Zohar[1], and Anne-Sophie Jannot[1,5]

[1]*INSERM, UMRS 1138, Centre de Recherche des Cordeliers, Paris, France*
[2]*LOPF, Califrais' Machine Learning Lab, Paris, France*
[3]*LPSM, UMR 8001, CNRS, Sorbonne University, Paris, France*
[4]*INSERM, UMR 1219, Bordeaux Population Health Research Center, Univ. Bordeaux, France*
[5]*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 both longitudinal and time-independent features are available. In the literature, standard joint models are either of type shared random-effect or joint latent class ones ; where the association structure between the longitudinal and the time-to-event submodels takes respectively the form of either shared association features learned from the longitudinal processes and included as potential risk factor in the survival model, or latent classes modeling population heterogeneity. We pick modeling ideas from both worlds and use appropriate penalties during inference for being able to learn from a high-dimensional context. The statistical performance of the method is examined on an extensive Monte Carlo simulation study, and finally illustrated on a publicly available dataset. Our proposed method significantly outperforms the state-of-the-art joint models regarding risk prediction in terms of C-index in a so-called real-time prediction paradigm, 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

*Corresponding author: simon.bussy@gmail.com

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 through a joint likelihood model rather than separately, has received considerable attention during the past two decades [Tsiatis and Davidian, 2004]. Numerical studies suggest that these approaches are among the most satisfactory to combine information [Yu et al., 2004]. 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) [Wulfsohn and Tsiatis, 1997], 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) [Vermunt and Magidson, 2003], in which the population is considered as heterogeneous, with the assumption that there exist homogeneous latent classes 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 joint models separately each one with univariate longitudinal outcome [Wang et al., 2012], which is inefficient [Lin et al., 2002a]. 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 Hickey et al. [2016] for a complete review). For instance, Jaffa et al. [2014] 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 trajectories, or longitudinal features, measured with error in a survival model. Let us consider the usual survival analysis framework. Following Andersen et al. [2012], 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 the time when a client stops using a company's product or service in a churn prediction setting ; or survival time, re-hospitalization, relapse or disease progression in a medical context. 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 [Bussy et al., 2019a]. To take account of this, we introduce a latent variable $G \in \{0, \dots, K-1\}$ modeling the $K \geq 1$ classes of different risk, which is a classical modeling assumption in JLCMs [Lin et al., 2002b, Proust-Lima et al., 2014]. 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 latent class, 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 inspired by 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 class-specific Cox model [Cox, 1972] 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 [Zou and Hastie, 2005] for time-independent feature selection in the latent class membership, and sparse group lasso [Simon et al., 2013] in the survival model : this penalty acts like the lasso at the trajectory level, namely an entire trajectory may drop out of the model, and yields sparsity for a given trajectory. Inference is achieved using an efficient and novel proximal Quasi-Newton Expectation Maximization algorithm (prox-QNEM). 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. Hence, the method provides interpretations of the high-dimensional longitudinal features, thus offering a powerful tool for real-time decision support in a customer's satisfaction monitoring context for instance, or else for clinical decision making in patient monitoring.

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 a publicly available dataset. 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 Class-specific marker trajectories

We suppose a class-specific marker trajectory and a generalized linear mixed model for each longitudinal marker given latent class 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 $r_l \leq q_l$. A suitable distributional assumption for the random effects component is a zero-mean multivariate normal distribution [Hickey et al., 2016], 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, which is not class-specific.

2.2 Class-specific risk of event

To quantify the effect of the longitudinal outcomes on the risk for an event, we use a Cox [Cox, 1972] relative risk model of the form

$$\lambda(t|\mathcal{Y}(t), G = k) = \lambda_0(t) \exp\Big\{ \sum_{l=1}^{L} \sum_{a=1}^{\mathcal{A}} \gamma_{k,a}^l \Psi_a^l(t) \Big\}, \tag{5}$$

where $\lambda_0(\cdot)$ is an unspecified baseline hazard function that does not depend on k, and we denote

$$\mathcal{Y}^l(t) = \{Y^l(u), 0 \leq u < t\} \quad \text{and} \quad \mathcal{Y}(t) = \bigcup_{l=1}^{L} \mathcal{Y}^l(t)$$

the history of the true longitudinal process up to time t. Then for each l-th longitudinal outcome, we consider $\mathcal{A} \in \mathbb{N}_+$ known functionals Ψ_a^l extracted from $\mathcal{Y}^l(t)$ through a given representation mapping (also independent from k), and $\gamma_{k,a}^l \in \mathbb{R}$ the corresponding joint representation parameters, being the only class-specific object in the risk of event submodel.

Let us finally denote

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

so that

$$\lambda(t|\mathcal{Y}(t), G = k) = \lambda_0(t) \exp\left\{\gamma_k^\top \Psi(t)\right\}, \tag{6}$$

where $\Psi(t) \in \mathbb{R}^{L\mathcal{A}}$ denotes the representation features vector.

2.3 Generalization of SREMs and JLCMs

Our model is clearly of JLCMs type, and can be viewed as a generalization of SREMs type in a much more flexible way. Table 1 gives common parameterizations for shared associations in SREMs. It is now common to include several of these in the survival sub-model, see for instance Andrinopoulou and Rizopoulos [2016], Ferrer et al. [2016], but as far as we know, there is no method to consider simultaneously a huge number of associations and let the model learn which one is predictive for the underlying task it is trained for.

Description	$\Psi_a^l(t, \beta_k^l, b^l)$	Reference
Linear predictor	$m_k^l(t)$	Chi and Ibrahim [2006]
Random effects	b^l	Hatfield et al. [2011]
Time-dependent slope	$\frac{\mathrm{d}}{\mathrm{d}t} m_k^l(t)$	Rizopoulos and Ghosh [2011]
Cumulative effect	$\int_0^t m_k^l(s)\mathrm{d}s$	Andrinopoulou et al. [2017]

Table 1: Examples of classical shared associations in SREMs.

Our proposition differs from standard SREMs in two respects: (i) representation vector $\Psi(t)$ does not depend on the modeling assumptions in the longitudinal submodel and (ii) it allows to choose multiple high-dimensional representation mappings that characterize time series and concatenate them, since we perform features selection through the regularization strategy described in Section 3.1. Point (i) is key since for instance the β_k^l dependence in Ψ_a^l would lead to complicated updates, while the b^l dependence would lead to untracktable integrals that are common in SREMs and require approximation methods such as Monte Carlo [Hickey et al., 2018], being computationally intensive and not scalable in a high-dimensional context. For point (ii) in practice, we use the `Python` library `tsfresh` [Christ et al., 2018] and include many extracted features, such as absolute energy of the time series, statistics on autocorrelation, or Fourier and wavelet basis projections, to name but a few.

2.4 Graphical model

Figure 1 below gives the graphical representation for the lights model to get a beter view of the dependence structure for the model.

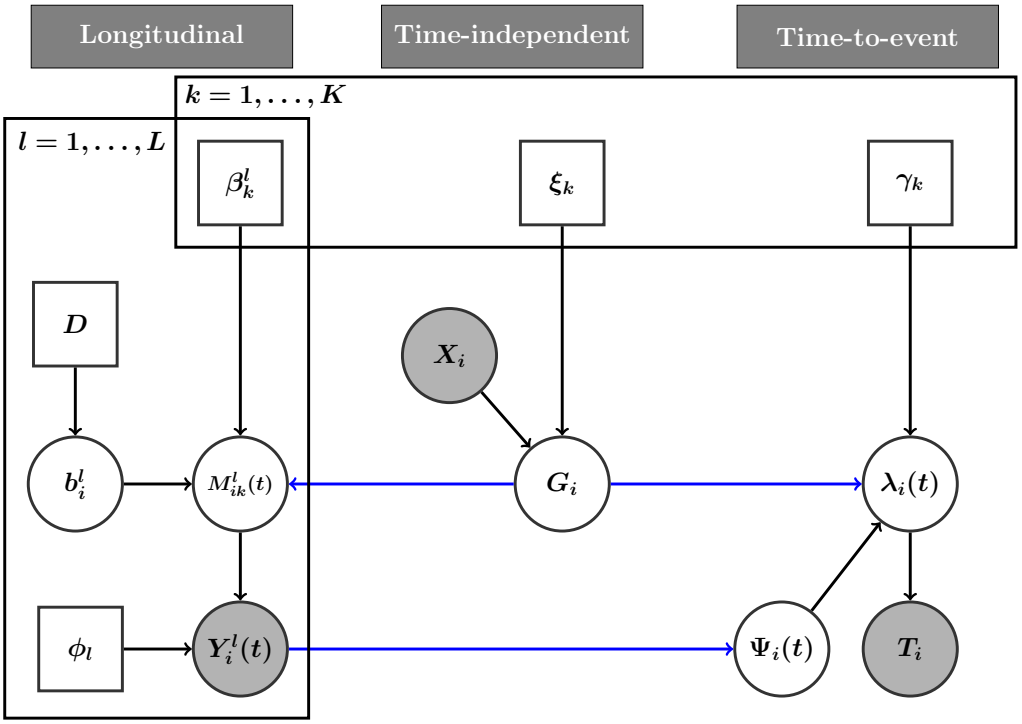

Figure 1: Graphical representation of the lights model, with ϕ_l being defined in (10) below. Square boxes denote observed data while circles denote unobserved (including random) terms. One can then observe the two links in blue between the submodels that gives rise to the joint model.

2.5 Likelihood

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

$$\mathcal{D}_n = \left\{(x_1, y_1^1, \ldots, y_1^L, t_1, \delta_1), \ldots, (x_n, y_n^1, \ldots, y_n^L, t_n, \delta_n)\right\} \tag{7}$$

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

$$y_i^l = (y_{i1}^l, \ldots, y_{in_i^l}^l)^\top \in \mathbb{R}^{n_i^l} \quad \text{with} \quad y_{ij}^l = Y_i^l(t_{ij}^l) \tag{8}$$

for all $l = 1, \ldots, L$. 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 $(V_i)_{i=1,\dots,n}$. Denoting $(t_1^u, \dots, t_J^u)$ the $J \in \mathbb{N}_+$ unique failure times, we then extract all representation vectors $\left(\Psi_i(t_j^u)\right)_{j=1,\dots,J}$ for all subjects $i = 1, \dots, n$ before inference and from now on, all computations are also done conditionally on them. Now, let us make two first hypothesis.

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

Assumption 1 states that subject-and-longitudinal outcome specific random effects b^l do not depend on the latent class membership, which is not a strong modeling assumption.

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

Assumption 2 is similar to standard modeling hypothesis, see for instance Tsiatis and Davidian [2004]. 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 case). 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

$$f(y_i|b_i, G_i = k) = \exp\left\{(y_i \odot \Phi_i)^\top M_{ik} - c_\phi(M_{ik}) + d_\phi(y_i)\right\}, \tag{9}$$

with

$$\Phi_i = (\phi_1^{-1}{\mathbf{1}_{n_i^1}}^\top \cdots \phi_L^{-1}{\mathbf{1}_{n_i^L}}^\top)^\top \in \mathbb{R}^{n_i} \tag{10}$$

and $\phi = (\phi_1, \dots, \phi_L)^\top \in \mathbb{R}^L$. The density described in (9) encompasses several distributions, see Table 2. The functions $c_\phi(\cdot)$ and $d_\phi(\cdot)$ are known as well as the dispersion parameter ϕ_l, while parameter β_k has to be estimated. Note that $d_\phi(\cdot)$ is related to the normalizing constant.

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

Let us denote

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

the collection of the $\vartheta \in \mathbb{N}_+$ unknown parameters to estimate. Note that we include λ_0 in θ. Indeed, we will estimate the baseline using nonparametric maximum likelihood, as a function taking mass at each failure time. So in θ, λ_0 represents the vector of these baseline hazard quantities, with a dimension equal to J. To write the log-likelihood $\ell_n(\theta)$ (rescaled by n^{-1}) for samples in $\mathcal{D}_n$ (defined in (7)), corresponding to the joint distribution of the time-to-event and longitudinal outcomes, let us make the following hypothesis.

Assumption 3. *Assume that*

$$f_\theta(t_i, \delta_i, y_i | b_i, G_i = k) = f_\theta(t_i, \delta_i | G_i = k) f_\theta(y_i | b_i, G_i = k) \tag{11}$$

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

Assumption 3 is a generalization of classical hypothesis used in SREMs and JLCMs [Hickey et al., 2016]. Remember that the densities are also conditionned by all the representation features $\left(\Psi_i(t_j^u)\right)_{j=1,\ldots,J}$. Hence Assumption 3 says that the random effects vector b_i, the latent class membership and the representation features account for the association between the longitudinal and event outcomes, which makes sense given Figure 1. Then, one has

$$\ell_n(\theta) = \ell_n(\theta; \mathcal{D}_n) = n^{-1} \sum_{i=1}^{n} \log \sum_{k=0}^{K-1} \pi_{\xi_k}(x_i) f_\theta(t_i, \delta_i | G_i = k) f_\theta(y_i | G_i = k), \tag{12}$$

where

$$f_\theta(t_i, \delta_i | G_i = k) = \left[\lambda(t_i | \mathcal{Y}_i(t_i), G_i = k)\right]^{\delta_i} S_k(t_i),$$

and

$$S_k(t_i) = \exp\left\{ - \int_0^{t_i} \lambda(s | \mathcal{Y}_i(s), G_i = k) \mathrm{d}s \right\} \tag{13}$$

is the survival probability of subject i given that it belongs to latent class k. As expected, (12) is a JLCMs like log-likelihood. Note that to write (12), one could simplify Assumption 3 by removing the b_i both in the left and right side of Equation (11) to get a classical JLCMs hypothesis, but the b_i mention is useful later on in the E-step of the inference algorithm, for instance to compute Equation (19).

Specification of the design matrices. In many practical applications, subjects show highly nonlinear longitudinal trajectories. We consider a flexible representations for $u^l(t)$ using a high-dimensional vector of time monomials, namely

$$u^l(t) = (1, t, t^2, \ldots, t^\alpha)^\top \tag{14}$$

with $\alpha \in \mathbb{N}_+$. The idea here is to let the practitioner choose a suitable polynomial order α for the representation – that could also be tuned automatically using a model selection procedure and AIC [Akaike, 1974] for instance. 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 $l = 1, \ldots, L$.

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, then focus on the algorithm proposed for inference, and on its convergence properties.

3.1 Penalized objective

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

$$\ell_n^{\text{pen}}(\theta) = -\ell_n(\theta) + \sum_{k=0}^{K-1} \zeta_{1,k}\|\xi_k\|_{\text{en},\eta} + \zeta_{2,k}\|\gamma_k\|_{\text{sgl}_1,\tilde{\eta}} \tag{15}$$

where for all $k = 0, \ldots, K-1$, we add an elastic net regularization [Zou and Hastie, 2005] of the vector ξ_k and a sparse group lasso regularization [Simon et al., 2013] of the vector γ_k, for tuning hyper-parameters $(\zeta_{1,k}, \zeta_{2,k})^\top \in \mathbb{R}_+^2$. Here, for fixed $(\eta, \tilde{\eta}) \in [0,1]^2$, one has

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

for the elastic net penalty of 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 of any vector z, with $\|z\|_{2,l} = \|z^l\|_2$, and 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

$$\hat{\theta} \in \text{argmin}_{\theta \in \mathbb{R}^\vartheta}\, \ell_n^{\text{pen}}(\theta). \tag{16}$$

One advantage of the considered regularization strategy is its ability to perform feature selection and pinpoint the most important features (trajectories and time-independent) relatively to the prediction objective. The support of $\hat{\xi}_k$ informs on the time-independent features involved in the k-th latent class membership. The ridge part allows to handle potential correlation between time-independent features. On the other hand, for the sparse group lasso penalty, a group l corresponds to a trajectory, that is one longitudinal outcome. So if $\hat{\gamma}_k^l$ is entirely zeroed (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 to perform representation features selection for each trajectory : for $\hat{\gamma}_k^l$ that are not entirely zeroed, its support informs on the representation features involved in the k-th latent class risk of event for the l-th longitudinal outcome. Note that in practice, the intercept terms are not regularized.

3.2 A Proximal Quasi-Newton EM

In order to derive an algorithm for this objective, we introduce a so-called prox-QNEM algorithm, being a combination between an EM algorithm [Dempster et al., 1977], an L-BFGS-B algorithm [Zhu et al., 1997] and a proximal gradient descent algorithm [Beck and Teboulle, 2009]. EM algorithm has already been used for multivariate data joint modeling

(see Lin et al. [2002a] for instance), but here we face different and 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 $\mathcal{D}_n$, $\boldsymbol{b} = (b_1^\top, \ldots, b_n^\top)$ and $\boldsymbol{G} = (G_1, \ldots, G_n)$. It can be written

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

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

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

$$\mathcal{Q}_n(\theta, \theta^{(w)}) = \mathbb{E}_{\theta^{(w)}}[\ell_n^{\text{comp}}(\theta) | \mathcal{D}_n]. \tag{17}$$

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

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

for different functions g, where we denote

$$\pi_{ik}^{\theta^{(w)}} = \mathbb{P}_{\theta^{(w)}}[G_i = k | t_i, \delta_i, y_i] \tag{18}$$

the posterior probability of the latent class membership using parameters $\theta^{(w)}$. Then, either $g(b_i, G_i) = \tilde{g}(b_i)$ (either $\tilde{g}_1 : b_i \mapsto b_i$ or $\tilde{g}_2 : b_i \mapsto b_i b_i^\top$), or $g(b_i, G_i) = g_k(G_i)$ with $g_k : G_i \mapsto \mathbb{1}_{\{G_i=k\}}$.

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

$$\mathbb{E}_{\theta^{(w)}}[\tilde{g}(b_i) | t_i, \delta_i, y_i] = \int_{\mathbb{R}^r} \tilde{g}(b_i) f_{\theta^{(w)}}(b_i | t_i, \delta_i, y_i) \mathrm{d}b_i = \frac{\sum_{k=0}^{K-1} \pi_{\xi_k^{(w)}}(x_i) \Lambda_{ik,\tilde{g}}^{\theta^{(w)}}}{\sum_{k=0}^{K-1} \pi_{\xi_k^{(w)}}(x_i) \Lambda_{ik,1}^{\theta^{(w)}}} \tag{19}$$

where

$$\Lambda_{ik,\tilde{g}}^{\theta^{(w)}} = f_{\theta^{(w)}}(t_i, \delta_i | G_i = k) f_{\theta^{(w)}}(y_i | G_i = k) \int_{\mathbb{R}^r} \tilde{g}(b_i) f_{\theta^{(w)}}(b_i | y_i, G_i = k) \mathrm{d}b_i, \tag{20}$$

and $\Lambda_{ik,1}^{\theta^{(w)}}$ obtained from $\Lambda_{ik,\tilde{g}}^{\theta^{(w)}}$ taking $\tilde{g} : b_i \mapsto 1$. In the Gaussian case used in practice one has

$$\begin{aligned}
y_i | G_i = k; \theta^{(w)} &\sim \mathcal{N}\big(U_i \beta_k^{(w)}, V_i D^{(w)} V_i^\top + \Sigma_i^{(w)}\big), \\
b_i | y_i, G_i = k; \theta^{(w)} &\sim \mathcal{N}\big(O_{i,k}^{(w)}, W_i^{(w)}\big),
\end{aligned}$$

with $O_{i,k}^{(w)} = W_i^{(w)} V_i^T {\Sigma_i^{(w)}}^{-1}(y_i - U_i \beta_k^{(w)})$ and $W_i^{(w)} = ({V_i}^\top {\Sigma_i^{(w)}}^{-1} V_i + {D^{(w)}}^{-1})^{-1}$. Hence, what is interesting here is that (20) is actually closed-formed since

$$\mathbb{E}_{\theta^{(w)}}[\tilde{g}_1(b_i)|y_i, G_i = k] = O_{i,k}^{(w)} \quad \text{and} \quad \mathbb{E}_{\theta^{(w)}}[\tilde{g}_2(b_i)|y_i, G_i = k] = W_i^{(w)} + O_{i,k}^{(w)} {O_{i,k}^{(w)}}^\top$$

and then tractable analytically. Whereas in classical SREMs, some form of approximation must be used in practice to compute this quantity, like numerical integration techniques (for instance Gaussian quadrature) or Monte Carlo approximation.

Now in the case $g = g_k$, one has

$$\mathbb{E}_{\theta^{(w)}}[g_k(Gi)|t_i, \delta_i, y_i] = \pi_{ik}^{\theta^{(w)}} = \frac{\pi_{\xi_k^{(w)}}(x_i)\Lambda_{ik,1}^{\theta^{(w)}}}{\sum_{k=0}^{K-1} \pi_{\xi_k^{(w)}}(x_i)\Lambda_{ik,1}^{\theta^{(w)}}}. \tag{21}$$

Proximal Quasi-Newton M-step. Let us denote

$$\mathcal{Q}_n^{\text{pen}}(\theta, \theta^{(w)}) = \mathcal{Q}_n(\theta, \theta^{(w)}) + \sum_{k=0}^{K-1} \zeta_{1,k}\|\xi_k\|_{\text{en},\eta} + \zeta_{2,k}\|\gamma_k\|_{\text{sgl}_1,\tilde{\eta}}. \tag{22}$$

Here, we need to compute

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

We then present the parameters updates in the order given in Algorithm 2. First, the update of $D^{(w)}$ is naturally given in closed-form by

$$D^{(w+1)} = n^{-1} \sum_{i=1}^{n} \mathbb{E}_{\theta^{(w)}}[b_i b_i^\top | t_i, \delta_i, y_i]. \tag{23}$$

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

$$P_{n,k}^{(w)}(\xi_k) = -n^{-1} \sum_{i=1}^{n} \sum_{k=0}^{K-1} \pi_{ik}^{\theta^{(w)}} \log \pi_{\xi_k}(x_i)$$

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

$$\xi_k^{(w+1)} \in \text{argmin}_{\xi_k \in \mathbb{R}^p} P_{n,k}^{(w)}(\xi_k) + \zeta_{1,k}\|\xi_k\|_{\text{en},\eta}. \tag{24}$$

It looks like the logistic regression objective, where labels are not fixed but softly encoded by the expectation step (computation of $\pi_{ik}^{\theta^{(w)}}$ in (21)). We then choose to solve (24) 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 penalization, 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 (24) as the following differentiable problem with box constraints

$$\begin{aligned} &\text{minimize} && P_{n,k}^{(w)}(\xi_k^+ - \xi_k^-) + \zeta_{1,k}\big((1-\eta)\sum_{j=1}^{p}(\xi_{k,j}^+ + \xi_{k,j}^-) + \frac{\eta}{2}\|{\xi_k}^+ - {\xi_k}^-\|_2^2\big) \\ &\text{subject to} && \xi_{k,j}^+ \geq 0 \text{ and } \xi_{k,j}^- \geq 0 \text{ for } j = 1, \ldots, p \end{aligned} \tag{25}$$

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

$$\frac{\partial P_{n,k}^{(w)}(\xi_k)}{\partial \xi_k} = -n^{-1} \sum_{i=1}^{n} \big(\pi_{ik}^{\theta^{(w)}} - \pi_{\xi_k}(x_i)\big) x_i. \tag{26}$$

In practice, we use the `Python` solver `fmin_l_bfgs_b` from `scipy.optimize` [Virtanen et al., 2020]. Note that for computational reasons, it is not clever to use a too small solver tolerance in the early steps of the overall algorithm, where we call tolerance the value for which iterations stop when the stopping criterion is below it. To speed up the overall algorithm, we then increase the L-BFGS-B tolerance with the iterations w.

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

$$R_{n,k}^{(w)}(\beta_k) = -n^{-1} \sum_{i=1}^{n} \pi_{ik}^{\theta^{(w)}} \Big[(y_i \odot \Phi_i^{(w)})^\top \mathbb{E}_{\theta^{(w)}}[M_{ik} | t_i, \delta_i, y_i] - \mathbb{E}_{\theta^{(w)}}[c_{\phi^{(w)}}(M_{ik}) | t_i, \delta_i, y_i] \Big]$$

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

$$\beta_k^{(w+1)} \in \operatorname{argmin}_{\beta_k \in \mathbb{R}^q} R_{n,k}^{(w)}(\beta_k). \tag{27}$$

The gradient of $R_{n,k}^{(w)}$ for the Gaussian case is here given by

$$\frac{\partial R_{n,k}^{(w)}(\beta_k)}{\partial \beta_k} = -n^{-1} \sum_{i=1}^{n} \pi_{ik}^{\theta^{(w)}} \Big[U_i^\top \Sigma_i^{(w)} I_{n_i} \big(y_i - U_i \beta_k - V_i \mathbb{E}_{\theta^{(w)}}[b_i | t_i, \delta_i, y_i] \big) \Big]. \tag{28}$$

The closed-form update is then given by

$$\beta_k{}^{(w+1)} = \frac{\sum_{i=1}^{n} \pi_{ik}^{\theta^{(w)}} \Big[U_i^\top I_{n_i} (y_i - V_i \mathbb{E}_{\theta^{(w)}}[b_i | t_i, \delta_i, y_i]) \Big]}{\sum_{i=1}^{n} \pi_{ik}^{\theta^{(w)}} U_i^\top I_{n_i} U_i}. \tag{29}$$

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

$$Q_{n,k}^{(w)}(\gamma_k) = -n^{-1} \sum_{i=1}^{n} \pi_{ik}^{\theta^{(w)}} \Big[\delta_i \gamma_k^\top \Psi_i(t_i) - \sum_{j=1}^{J} \lambda_0^{(w)}(t_j^u) \exp\big\{ \gamma_k^\top \Psi_i(t_j^u) \big\} \mathbb{1}_{\{t_j^u \leq t_i\}} \Big]$$

based on the quantities involved in $\mathcal{Q}_n(\theta, \theta^{(w)})$ that depend on γ_k. The integration over the survival process has been replaced with a finite sum over the process evaluated at the J, since $\lambda_0^{(w)}$ defined in (33) is always zero except at observed event times. The update for $\gamma_k^{(w)}$ requires to solve the following convex minimization problem

$$\gamma_k^{(w+1)} \in \operatorname{argmin}_{\gamma_k \in \mathbb{R}^{L\mathcal{A}}} Q_{n,k}^{(w)}(\gamma_k) + \zeta_{2,k} \|\gamma_k\|_{\mathrm{sgl}_1, \tilde{\eta}}. \tag{30}$$

The previous trick we used to handle the non-differentiability of the ℓ_1 part in the elastic net is no longer sufficient for (30), 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 [Simon et al., 2013]). Then, we

choose to solve problem (30) using the proximal gradient descent [Boyd and Vandenberghe, 2004] described in Algorithm 1, where we first define

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

Algorithm 1: Proximal gradient descent $\text{ISTA}_{\text{sgl}_1}(v,\, f,\, \tilde{\eta},\, \zeta)$

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

Initialize $\flat^{(0)} = v$, $\varrho^0 > 0$, some $\epsilon > 1$
for $\breve{w} = 0, \ldots,$ *until convergence* **do**
  Backtracking line search : find the smallest nonnegative integer $i^{\breve{w}}$ such that
  $$f\left(p_{\tilde{\varrho}_{\breve{w}}}(\flat^{(\breve{w})})\right) \leq f(\flat^{(\breve{w})}) + \left\langle \left(p_{\tilde{\varrho}_{\breve{w}}}(\flat^{(\breve{w})}) - \flat^{(\breve{w})}\right), \nabla f(\flat^{(\breve{w})}) \right\rangle + \frac{1}{2\tilde{\varrho}_{\breve{w}}} \|p_{\tilde{\varrho}_{\breve{w}}}(\flat^{(\breve{w})}) - \flat^{(\breve{w})}\|_2^2$$
  with $\tilde{\varrho}_{\breve{w}} = \epsilon^{i^{\breve{w}}} \varrho^{\breve{w}}$
  Set $\varrho^{(\breve{w}+1)} = \tilde{\varrho}_{\breve{w}}$
  $\flat^{(\breve{w}+1)} = p_{\varrho^{(\breve{w}+1)}}(\flat^{(\breve{w})}) := \text{prox}_{\text{sgl}_1, \tilde{\eta}, \zeta}\left(\flat^{(\breve{w})} - \varrho^{(\breve{w}+1)} \nabla f(\flat^{(\breve{w})})\right)$
  if $\dfrac{\|\flat^{(\breve{w}+1)} - \flat^{(\breve{w})}\|_2}{\varrho^{(\breve{w}+1)}} < tol$ **then**
    Return: $\flat^{\breve{w}}$

Lemma 1 bellow states that one can easily compute the proximal operator of the sparse group lasso defined in Algorithm 1.

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 [Bach et al., 2011] and tractable. We used the `Python` library `copt` for the implementation of proximal gradient descent [Fabian Pedregosa, 2020] and we propose in our `lights` package a first `Python` 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 $\gamma_k^{(w)}$ with the following step

$$\gamma_k^{(w+1)} = \text{ISTA}_{\text{sgl}_1}\left(\gamma_k^{(w)}, Q_{n,k}^{(w)}(\cdot), \tilde{\eta}, \zeta_{2,k}\right) \tag{31}$$

Note that the warmstart initialization is left to the user's choice, as well as the ISTA acceleration [Beck and Teboulle, 2009], since warmstart can diminish acceleration effectiveness [Tibshirani, 2010]. The gradient of $Q_{n,k}^{(w)}$ is here given by

$$\frac{\partial Q_{n,k}^{(w)}(\gamma_k)}{\partial \gamma_{k,a}^{l}} = -n^{-1} \sum_{i=1}^{n} \pi_{ik}^{\theta^{(w)}} \Psi_{i,a}^{l}(t_i) \left[\delta_i - \sum_{j=1}^{J} \lambda_0^{(w)}(t_j^u) \exp\left\{ \gamma_k^{\top} \Psi_i(t_j^u) \right\} \mathbb{1}_{\{t_j^u \leq t_i\}} \right] \tag{32}$$

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 each uncensored observed event times as parameters to be estimated [Klein, 1992], and the closed-form update obtained by cancelling the gradient is given by

$$\lambda_0^{(w+1)}(t) = \frac{\sum_{i=1}^n \delta_i \mathbb{1}_{\{t=t_i\}}}{\sum_{i=1}^n \sum_{k=0}^{K-1} \pi_{ik}^{\theta^{(w)}} \exp\left\{\gamma_k^{(w+1)^\top} \Psi_i(t)\right\} \mathbb{1}_{\{t_i \geq t\}}}, \tag{33}$$

for all $t \geq 0$, which is a Breslow like estimator [Breslow, 1972] 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

$$\phi_l^{(w+1)} = \Big(\sum_{i=1}^n n_i^l\Big)^{-1} \sum_{i=1}^n \sum_{k=0}^{K-1} \pi_{ik}^{\theta^{(w)}} \Big[\big(y_i^l - U_{il}\beta_k^{l\,(w+1)}\big)^\top \Big(y_i^l - U_{il}\beta_k^{l\,(w+1)} - 2V_{il}\mathbb{E}_{\theta^{(w)}}[b_i^l | t_i, \delta_i, y_i]\Big) + \mathrm{Tr}\left(V_{il}^\top V_{il} \mathbb{E}_{\theta^{(w)}}[b_i^l b_i^{l\,\top} | t_i, \delta_i, y_i]\right)\Big]. \tag{34}$$

Appendix E.2 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 2. For all $k = 0, \ldots, K-1$, we first choose $\xi_k^{(0)} = \mathbf{0}_d$ and $\gamma_k{}^{(0)} = \mathbf{0}_{L\mathcal{A}}$. Then, we initialize $\lambda_0^{(0)}$ like if there is no latent classes ($\gamma_0^{(0)} = \cdots = \gamma_{K-1}^{(0)}$) 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 latent classes ($\beta_0^{(0)} = \cdots = \beta_{K-1}^{(0)}$) – using a multivariate linear mixed model with an explicit EM algorithm (details on the algorithm are given in Appendix D, and details on the implementation in Appendix E.1), being itself initialized with univariates fits using the `Python` package `statsmodels`.

The prox-QNEM algorithm. Algorithm 2 describes the main steps of the resulting prox-QNEM algorithm to solve the optimization problem (16).

Then in practice, the convergence for Algorithm 2 is declared when the relative objective $\big(\ell_n^{\mathrm{pen}}(\theta^{(w+1)}) - \ell_n^{\mathrm{pen}}(\hat{\theta}^{(w)})\big)/\ell_n^{\mathrm{pen}}(\hat{\theta}^{(w)})$ goes below a chosen threshold.

3.3 Convergence to a stationary point

Let us address here convergence properties of the prox-QNEM algorithm described in the previous section. Since in (22) we only minimize $\mathcal{Q}_n^{\mathrm{pen}}(\theta, \theta^{(w)})$ and not directly $\mathcal{Q}_n(\theta, \theta^{(w)})$ defined in (17) as in a classical EM [Wu, 1983], we are in a generalized EM (GEM) setting [Dempster et al., 1977]. For such algorithm, one has the descent property saying that the objective function (15) decreases at each iteration, namely

$$\ell_n^{\mathrm{pen}}(\theta^{(w+1)}) \leq \ell_n^{\mathrm{pen}}(\theta^{(w)}),$$

which can be observed in Figure ?. Now, let us make a classical hypothesis.

Assumption 4. *Suppose that ℓ_n^{pen} is bounded for all θ.*

Algorithm 2: prox-QNEM algorithm for the lights model inference

Input: Training data $\mathcal{D}_n$ (7); tuning hyper-parameters $(\zeta_{1,k}, \zeta_{2,k})_{k=0,\dots,K-1}$

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

/* Initialization */

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

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

  /* E-step */

  Compute $\Lambda_{ik,\tilde{g}}^{\theta^{(w)}}$ using (20) for $\tilde{g}_1$ and $\tilde{g}_2$

  Compute $\pi_{ik}^{\theta^{(w)}}$ using (21)

  /* Proximal Quasi-Newton M-step */

  Update $D^{(w+1)}$ using (23)

  Update $(\xi_k^{(w+1)})_{k=0,\dots,K-1}$ by solving (25)

  Update $(\beta_k^{(w+1)})_{k=0,\dots,K-1}$ by solving (29)

  Update $(\gamma_k^{(w+1)})_{k=0,\dots,K-1}$ by solving (31)

  Update $\lambda_0^{(w+1)}$ using (33)

  Update $\phi^{(w+1)}$ using (34)

Return: $\hat{\theta}$

Assumption 4 implies that $w \mapsto \ell_n^{\text{pen}}(\theta^{(w)})$ decreases monotically to a finite limit. Then, one has the following convexity property with the Gaussian case used in practice.

Lemma 2. *$\mathcal{Q}_n^{pen}(\theta, \theta^{(w)})$ is continuous in θ and $\theta^{(w)}$, and for any fixed $\theta^{(w)}$, it is a convex function in θ, and strictly convex in each coordinate of θ.*

Hence, one can state the following theorem about the convergence of the prox-QNEM algorithm to a stationary point, being here a local minimum.

Theorem 1. *Under Assumption 4, every cluster point $\bar{\theta}$ of the sequence $\{\theta^{(w)}\}_{w\geq 1}$ generated by the prox-QNEM algorithm is a stationary point of the objective function (15), that is*

$$\forall \varepsilon > 0, V_\varepsilon(\bar{\theta}) \cap \{\theta^{(w)}\}_{w\geq 1} \setminus \{\bar{\theta}\} \neq \varnothing \implies \forall \tilde{\theta} \in \mathbb{R}^{\vartheta}, \lim_{u \to 0} \frac{\ell_n^{pen}(\bar{\theta} + \tilde{\theta} u) - \ell_n^{pen}(\bar{\theta})}{u} \geq 0,$$

with $V_\varepsilon(\bar{\theta})$ the epsilon-neighbourhood of $\bar{\theta}$.

The proof of Lemma 2 is straightforward, and we follow the proof of Theorem 1 in Bussy et al. [2019a] to prove Theorem 1.

3.4 The case $K = 2$

In many practical applications – including those addressed in Section 6 – we are interested in identifying one latent class 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 latent class 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 γ_k does not depend on k, so that we denote $\zeta_2 = \zeta_{2,0} = \zeta_{2,1}$.

Cross-validation procedure. The hyper-parameters $(\zeta_1, \zeta_2)^\top \in \mathbb{R}_+^2$ are then tuned using a Bayesian optimization method [Snoek et al., 2012], with the C-index score and the predictive marker both defined in Section 4.1. In practice, we use the `Python` package `hyperopt` [Bergstra et al., 2013] and the Tree-structured Parzen Estimator [Bergstra et al., 2011], which is a sequential model-based optimization (SMBO) approach. SMBO methods sequentially construct models to approximate the performance of hyperparameters based on historical measurements, and then subsequently choose new hyperparameters to test based on this model. 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 is 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 considered models for performance comparisons.

4.1 Evaluation strategy

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 [Heagerty and Zheng, 2005] (defined in (36)), or in the predictive accuracy of the rule [Schemper and Henderson, 2000]. 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 [Hickey et al., 2016]. 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 what we call a "real-time prediction paradigm".

Real-time prediction paradigm. Once the learning phase is achieved for the model on a training set (so that one obtains $\hat{\theta}$ from (16) using our prox-QNEM algorithm introduced in Section 3.2), we want to assess real-time risk prediction performances on a test set. Denoting t_i^{max} the time for subject i in the test set 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 – one wants then to predict a prognostic marker for subject i using all data available up to t_i^{max}, but without using the supervision labels (t_i, δ_i).

Predictive marker. The question is now 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. Posterior risk classification for subject i and latent class k can be made through $\pi_{ik}^{\hat{\theta}}$ (see (21)), and such posterior probabilities have been considered for goodness-of-fit evaluation in other JLCMs models, see Proust-Lima et al. [2014] 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, since the latter requires to know the survival labels (t_i, δ_i), being intrinsically unknow in the paradigm. A natural and appropriate marker

rule of the lights model for subject i being in latent class k in our context would rather be

$$\hat{\mathcal{R}}_{ik} = \mathbb{P}_{\hat{\theta}}[G_i = k | T_i^\star > t_i^{max}, y_i] = \frac{\pi_{\hat{\xi}_k}(x_i) S_k(t_i^{max}) f_{\hat{\theta}}(y_i | G_i = k)}{\sum_{k=0}^{K-1} \pi_{\hat{\xi}_k}(x_i) S_k(t_i^{max}) f_{\hat{\theta}}(y_i | G_i = k)}, \tag{35}$$

where S_k is defined in (13). Note that marker (35) is obtained from $\pi_{ik}^{\hat{\theta}}$ in (21) taking $(t_i, \delta_i) = (t_i^{max}, 0)$. Let us recall that in practice, one considers the case $K = 2$ for the lights, see Section 3.4.

The C-index metric We detail in this section the chosen metric to evaluate risk prediction performances. Let us denote by $\hat{\mathcal{R}}_i$ the predictive marker for a subject i at t_i^{max} and from a given model (for the lights, $\hat{\mathcal{R}}_i = \hat{\mathcal{R}}_{ik}$ in (35) with $k = 1$). A common concordance measure that does not depend on time is the C-index [Harrell et al., 1996] 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

$$\mathcal{C}_\tau = \mathbb{P}[\hat{\mathcal{R}}_i > \hat{\mathcal{R}}_j | T_i < T_j, T_i < \tau], \tag{36}$$

with τ corresponding to the fixed and prespecified follow-up period duration [Heagerty and Zheng, 2005]. A Kaplan-Meier estimator for the censoring distribution leads to a nonparametric and consistent estimator of $\mathcal{C}_\tau$ [Uno et al., 2011], 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 Blanche et al. [2015] 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_i^{max}. But comparing curves in an automatic way (in a cross-validation process for instance) is complicated, and 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 -v[Bussy et al., 2019b]), which is not the case of the C-index metric. Finally, the question of the choice of t_i^{max} in a study to assess a model performances, or to compare models, is discussed in the following paragraph. It can be sensed that the closer t_i^{max} gets to $T_i^\star$, the easier it will be for a model to make good predictions.

Procedure. Let us describe now in Algorithm 3 the procedure we follow to evaluate model performances on simulated data in Section 5 and on real data in Section 6. Here we do not give details on the cross-validation step, but it follows the same idea : for each fold, we sample the t_i^{max} and truncate y_i for i in the test set of the fold.

4.2 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.

Algorithm 3: Procedure followed to assess performances of a given model in our real-time prediction paradigm.

Input: Dataset $\mathcal{D}_n$ (7) **only if** *simulation = false* ; a `model` under study
Output: Confidence intervals on C-index metric as well as on running time.

/* We run $\breve{R} = 100$ independant experiments */
$seed_0 = 0$
for $\breve{r} = 1, \ldots, \breve{R}$ **do**
  `start_time` $=$ `time`()
  $seed_{\breve{r}} = seed_{\breve{r}-1} + 1$
  if *simulation = true* **then**
    $\mathcal{D}_n \leftarrow$ `simulate`$(seed_{\breve{r}})$ /* Sample a new dataset at each round $\breve{r}$ */
  $(\mathcal{D}_{n_{train}}, \mathcal{D}_{n_{test}}) \leftarrow$ `split_train_test`$(\mathcal{D}_n, seed_{\breve{r}})$
  `model.cross_validate`$(\mathcal{D}_{n_{train}})$ /* Select the best hyper parameters */
  `model.fit`$(\mathcal{D}_{n_{train}})$ /* Learn $\hat{\theta}$ */
  for $i = 1, \ldots, n_{test}$ **do**
    $t_i^{max} \sim t_i \times \big(1 - \text{Beta}(2, 5)\big)$ /* See Figure 2 */
    $y_i \leftarrow \big(Y_i^l(t_{ij}^l)\big)_{\substack{j=1,\ldots,n_i^l-1 \\ l=1,\ldots,L}}$ with $t_{ij}^l \leq t_i^{max}$ /* Truncate y_i, see (8) */
    $\mathcal{X}_i = (x_i, y_i)$
    if `model = lights` **then**
      $\big(\Psi_i(t_j^u)\big)_{j=1,\ldots,J} \leftarrow$ `extract_features`(y_i) /* $(t_j^u)_{j=1,\ldots,J}$ from $\mathcal{D}_{n_{train}}$ */
      $\mathcal{X}_i = \Big(x_i, y_i, \big(\Psi_i(t_j^u)\big)_{j=1,\ldots,J}\Big)$
    $\hat{\mathcal{R}}_{i,\breve{r}} \leftarrow$ `model.predict`$(\mathcal{X}_i)$
  `score`$_{\breve{r}} \leftarrow$ `c_index`$\big((\hat{\mathcal{R}}_{i,\breve{r}}, t_i, \delta_i)_{i=1,\ldots,n_{test}}\big)$
  `end_time` $=$ `time`()
  `running_time`$_{\breve{r}}$ $=$ `end_time` $-$ `start_time`
Return: $\big[\texttt{score}_{\breve{r}}\big]_{\breve{r}=1,\ldots,\breve{R}}$, $\big[\texttt{running\_time}_{\breve{r}}\big]_{\breve{r}=1,\ldots,\breve{R}}$

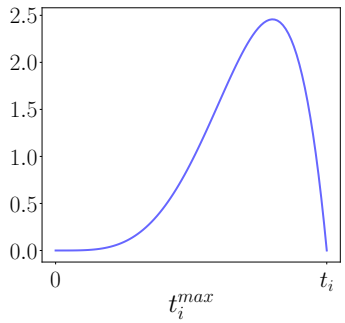

Figure 2: Density of the law used to simulate t_i^{max} to mimic the fact that in practice, one has access to a reasonable amount of longitudinal data before making a prediction.

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 Cox [1972], 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 [Therneau and Grambsch, 2000], 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 [Breslow, 1972].

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` [Christ et al., 2018]. 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 [Sueyoshi, 1992] (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 [Zhang et al., 2018], 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` [Proust-Lima et al., 2017]. There is no shared associations between the longitudinal models and the survival model. Given the latent class 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 [Rizopoulos, 2016]. The predictive marker associated with this model for subject i is naturally given by the corresponding risk score

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

where the shared associations $\varphi_{k,a}$ are the four given in Table 1.

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 3.4). We first choose a coefficient vector

$$\xi = (\underbrace{\varsigma_1, \dots, \varsigma_1}_{s}, 0, \dots, 0) \in \mathbb{R}^p, \tag{37}$$

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 [Mukherjee and Maiti, 1988] 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 trajectories start on average higher for high-risk subjects ($G_i = 1$), and with a higher slope than low-risk subjects ($G_i = 0$). This choice makes sens regarding that for simplicity, we take joint association parameters being all positive, see (38).

Most joint modeling simulation studies use fixed time points [Hickey et al., 2018], 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 [Hawkes and Oakes, 1974] 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 [Bacry et al., 2017]. An example of simulated longitudinal features for an individual is displayed in Figure 3(b) below.

Now to generate survival times, we want to induce sparsity in the longitudinal data effects, so we set

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

and we choose a risk model (5) with

$$\begin{cases} \big(\varphi_{k,a}(t, b_i^l)\big)_{a\in\{1,2\}} = \big(\beta_{k,1}^l + \beta_{k,2}^l t + b_{i,1}^l + b_{i,2}^l t \ , {b_i^l}^\top\big)^\top \mathbb{1}_{\{l\in\mathcal{S}_k\}} + \mathcal{N}\big(0, \Sigma_3(\rho_3)\big)\mathbb{1}_{\{l\notin\mathcal{S}_k\}} \\ \big(\varphi_{k,a}(t, b_i^l)\big)_{a=3,\dots,\mathcal{A}} = \mathcal{N}\big(0, \Sigma_4(\rho_3)\big), \end{cases}$$

rmq : plus besoin du sk dans phi vu qu'on l'ajoute directement ds le Y non ?
and

$$\gamma_{k,a}^l = \varsigma_2 \mathbf{1}_{\iota_a} \mathbb{1}_{\{l\in\mathcal{S}_k,\ a\in\{1,2\}\}} \tag{38}$$

for the choice of the association parameters, with $\mathcal{A} = 3 + 3(1 - r_s)/r_s$ so that (i) there is a proportion r_s of active longitudinal features among the L (with the $\mathcal{S}_k$), (ii) we induce sparsity within each latent class adding a proportion of $(1 - r_s)$ non-active association features ($a \geq 3$) in each latent class (similarly to (37) for the time-independent features), and (iii) the longitudinal effects are distinct for each latent class $k = 0, \dots, K - 1$ (thanks to the $\mathcal{S}_k$ definition).

In terms of active association features, it corresponds to (5) taking the first two functionals described in Table 1 as shared associations with $\alpha = 1$ in (14) so 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 that allows the following explicit survival times generation process. Then, we do not consider the time-dependent slope as association feature since here with $\alpha = 1$, it would be collinear with the random effects. We choose a Gompertz distribution [Gompertz, 1825] for the baseline, that is

$$\lambda_0(t) = \kappa_1 \kappa_2 \exp(\kappa_2 t) \tag{39}$$

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. One can now generate survival times explicitly as

$$T_i^\star | G_i = k \sim \frac{1}{\iota_{i,k,2} + \kappa_2} \log\left(1 - \frac{(\iota_{i,k,2} + \kappa_2)\log U_i}{\kappa_1 \kappa_2 \exp \iota_{i,k,1}}\right) \tag{40}$$

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 $r_c \in [0,1]$. The choice of all hyper-parameters is driven by the applications on real data presented in Section 6, and summarized in Table 3. Figure 3(b) gives an example of data generated according to the design we have just described.

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	(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	$(10^{-3}, 0.1)$	(1, 0.1)	0.3	0.4	0.1

(a) blablaex-ample simu (b)

Figure 3: Illustrations of simulated data.

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 4.2 in terms of C-index (see Section 4.1). 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 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 features and the trajectories 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 [Fleming and Harrington, 2011, Murtaugh et al., 1994]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 [Johnson et al., 2016]. 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 (prox-QNEM) 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

The proximal operator [Moreau, 1962] for any function $f : \mathbb{R}^p \to \mathbb{R}$ is defined by

$$\text{prox}_f(u) = \text{argmin}_{x\in\mathbb{R}^p} \{\frac{1}{2}\|x-u\|_2^2 + f(x)\}. \tag{41}$$

To prove Lemma 1, we need to show that

$$\text{prox}_{\lambda_1\|\cdot\|_1+\lambda_2\sum_{l=1}^L \|\cdot\|_{2,l}} = \text{prox}_{\lambda_2\sum_{l=1}^L \|\cdot\|_{2,l}} \circ \text{prox}_{\lambda_1\|\cdot\|_1}$$

where $\text{prox}_{\lambda_1\|\cdot\|_1+\lambda_2\sum_{l=1}^L \|\cdot\|_{2,l}}$ is the proximal operator for the sparse group lasso regularizer, $\text{prox}_{\lambda_2\sum_{l=1}^L \|\cdot\|_{2,l}}$ is the proximal operator for the group lasso regularizer and $\text{prox}_{\lambda_1\|\cdot\|_1}$ is the proximal operator for the lasso regularizer. $\lambda_1 > 0$ and $\lambda_2 > 0$ are regularization parameters, and $\|\cdot\|_{2,l}$ denotes the Euclidean norm on features in the l-th group.

Proof. Fix any $\beta \in \mathbb{R}^p$ and denote

$$k_\beta(x) = \frac{1}{2}\|x - \text{prox}_{\lambda_1\|\cdot\|_1}(\beta)\|_2^2 + \lambda_2 \sum_{l=1}^L \|x\|_{2,l}$$

for all $x \in \mathbb{R}^p$. Given that $\big(\text{prox}_{\lambda_1\|\cdot\|_1}(\beta)\big)_j = \max(0, 1-\lambda_1/|\beta_j|)\beta_j$ [Bach et al., 2011], k_β is clearly strictly convex. Denoting $x^\star$ its unique minimizer, (41) entails

$$\text{prox}_{\lambda_2\sum_{l=1}^L \|\cdot\|_{2,l}}\big(\text{prox}_{\lambda_1\|\cdot\|_1}(\beta)\big) = \text{argmin}_{x\in\mathbb{R}^p}\, k_\beta(x) = \{x^\star\}.$$

Then, it follows that

$$0 \in \partial k_\beta(x^\star) = x^\star - \text{prox}_{\lambda_1\|\cdot\|_1}(\beta) + \partial\big(\lambda_2 \sum_{l=1}^L \|x^\star\|_{2,l}\big) \tag{42}$$

where ∂k denotes the subgradient [Rockafellar, 1974] of k. Now denoting

$$h_\beta(x) = \frac{1}{2}\|x-\beta\|_2^2 + \lambda_1\|x\|_1 + \lambda_2 \sum_{l=1}^L \|x\|_{2,l}$$

for all $x \in \mathbb{R}^p$ such that

$$\text{prox}_{\lambda_1\|\cdot\|_1+\lambda_2\sum_{l=1}^L \|\cdot\|_{2,l}}(\beta) = \text{argmin}_{x\in\mathbb{R}^p}\, h_\beta(x),$$

it remains to be proved that $0 \in \partial h_\beta(x^\star)$.

Then, it turns out that

$$\partial h_\beta(x) = x - \beta + \lambda_1 \text{SGN}(x) + \partial(\lambda_2 \sum_{l=1}^L \|x\|_{2,l})$$

for all $x \in \mathbb{R}^p$, where $\mathrm{SGN}(x)$ is the subgradient of the ℓ_1-norm at x [Bach et al., 2011] defined in a component wise fashion as

$$\mathrm{SGN}(x) = \begin{cases} \{1\} & \text{if } x > 0, \\ [\text{-}1,\ 1] & \text{if } x = 0, \\ \{-1\} & \text{if } x < 0. \end{cases}$$

Denoting $y = \mathrm{prox}_{\lambda_1\|\cdot\|_1}(\beta)$, Lemma 1.4) in Yuan et al. [2011] entails $\mathrm{SGN}(y) \subseteq \mathrm{SGN}(x^*)$. Hence,

$$y = \mathrm{argmin}_{u\in\mathbb{R}^p} \left\{ \frac{1}{2}\|u-\beta\|_2^2 + \lambda_1\|u\|_1 \right\} = \beta - \lambda_1 \mathrm{SGN}(y) \subseteq \beta - \lambda_1 \mathrm{SGN}(x^*),$$

which gives

$$0 \in x^\star - y + \partial(\lambda_2 \sum_{l=1}^{L} \|x^\star\|_{2,l}) \subseteq x^\star - \beta + \lambda_1 \mathrm{SGN}(x^\star) + \partial(\lambda_2 \sum_{l=1}^{L} \|x^\star\|_{2,l}).$$

□

B Automatic cross-validation grid for ξ

Concerning the cross-validation procedure for tuning

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

we use a randomized search with the C-index metric (see Section 4.1) where for all $k = 0, \ldots, K-1$ and $j \in \{1, 2\}$, 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 (24) 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 $j = 1, \ldots, p$ are exactly zero. The KKT conditions [Boyd and Vandenberghe, 2004] 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 $\hat{\mathcal{A}}_k = \{j = 1, \ldots, p : \hat{\xi}_{k,j} \neq 0\}$ 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 (26), one obtains

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

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

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

Similar strategy to automatically compute $\zeta_{2,k}^{\max}$ is not easy since without additionnal hypothesis, the gradients 28 and 32 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 (40) used to generate event times. Indeed, we can not refer to Austin [2012] since the intensity (39) of the baseline is not correct in this paper. Note that other errors had already been pointed in [Austin, 2013], 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 (39), 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

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

and one can generate event times according to

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

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 prox-QNEM algorithm in Section 3.2, acting as if there is no latent classes ($\beta_0^{(0)} = \cdots = \beta_{K-1}^{(0)}$). 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

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

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

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

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

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

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

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 QNEM 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

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

E.2 `prox-QNEM class`

Let us introduce the following useful notations

$$\begin{cases} \tilde{g}^1 & : b_i \mapsto b_i, \\ \tilde{g}^2 & : b_i \mapsto b_i b_i^\top. \end{cases}$$

References

Hirotugu Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974.

Per K Andersen, Ornulf Borgan, Richard D Gill, and Niels Keiding. *Statistical models based on counting processes*. Springer Science & Business Media, 2012.

Galen Andrew and Jianfeng Gao. Scalable training of l1-regularized log-linear models. In *International Conference on Machine Learning*, pages 33–40. ACM, 2007.

Eleni-Rosalina Andrinopoulou and Dimitris Rizopoulos. Bayesian shrinkage approach for a joint model of longitudinal and survival outcomes assuming different association structures. *Statistics in medicine*, 35(26):4813–4823, 2016.

Eleni-Rosalina Andrinopoulou, Dimitris Rizopoulos, Johanna JM Takkenberg, and Emmanuel Lesaffre. Combined dynamic predictions using joint models of two longitudinal outcomes and competing risk data. *Statistical methods in medical research*, 26(4):1787–1801, 2017.

Peter C Austin. Generating survival times to simulate cox proportional hazards models with time-varying covariates. *Statistics in medicine*, 31(29):3946–3958, 2012.

Peter C Austin. Correction:'generating survival times to simulate cox proportional hazards models with time-varying covariates'. *Statistics in Medicine*, 32(6):1078–1078, 2013.

Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Optimization with sparsity-inducing penalties. *arXiv preprint arXiv:1108.0775*, 2011.

Emmanuel Bacry, Martin Bompaire, Philip Deegan, Stéphane Gaïffas, and Søren V Poulsen. Tick: a python library for statistical learning, with an emphasis on hawkes processes and time-dependent models. *The Journal of Machine Learning Research*, 18 (1):7937–7941, 2017.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.

James Bergstra, Dan Yamins, David D Cox, et al. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In *Proceedings of the 12th Python in science conference*, volume 13, page 20. Citeseer, 2013.

Paul Blanche, Cécile Proust-Lima, Lucie Loubère, Claudine Berr, Jean-François Dartigues, and Hélène Jacqmin-Gadda. Quantifying and comparing dynamic predictive accuracy of joint models for longitudinal marker and time-to-event in presence of censoring and competing risks. *Biometrics*, 71(1):102–113, 2015.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Norman E Breslow. Contribution to discussion of paper by dr cox. *JJournal of the Royal Statistical Society: Series B (Statistical Methodology)*, 34:216–217, 1972.

Simon Bussy, Agathe Guilloux, Stéphane Gaïffas, and Anne-Sophie Jannot. C-mix: A high-dimensional mixture model for censored durations, with applications to genetic data. *Statistical methods in medical research*, 28(5):1523–1539, 2019a.

Simon Bussy, Raphaël Veil, Vincent Looten, Anita Burgun, Stéphane Gaïffas, Agathe Guilloux, Brigitte Ranque, and Anne-Sophie Jannot. Comparison of methods for early-readmission prediction in a high-dimensional heterogeneous covariates and time-to-event outcome framework. *BMC medical research methodology*, 19(1):50, 2019b.

Yueh-Yun Chi and Joseph G Ibrahim. Joint models for multivariate longitudinal and multivariate survival data. *Biometrics*, 62(2):432–445, 2006.

Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W Kempa-Liehr. Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package). *Neurocomputing*, 307:72–77, 2018.

D. R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2):187–220, 1972.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

Gideon Dresdner Fabian Pedregosa, Geoffrey Negiar. copt: composite optimization in python. 2020. doi: 10.5281/zenodo.1283339. URL http://openopt.github.io/copt/.

Loïc Ferrer, Virginie Rondeau, James Dignam, Tom Pickles, Hélène Jacqmin-Gadda, and Cécile Proust-Lima. Joint modelling of longitudinal and multi-state processes: application to clinical progressions in prostate cancer. *Statistics in medicine*, 35(22):3933–3948, 2016.

Thomas R Fleming and David P Harrington. *Counting processes and survival analysis*, volume 169. John Wiley & Sons, 2011.

Benjamin Gompertz. 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. *Philosophical transactions of the Royal Society of London*, (115):513–583, 1825.

Frank E Harrell, Kerry L Lee, and Daniel B Mark. Tutorial in biostatistics multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in medicine*, 15:361–387, 1996.

Laura A Hatfield, Mark E Boye, and Bradley P Carlin. Joint modeling of multiple longitudinal patient-reported outcomes and survival. *Journal of Biopharmaceutical Statistics*, 21(5):971–991, 2011.

Alan G Hawkes and David Oakes. A cluster process representation of a self-exciting process. *Journal of Applied Probability*, 11(3):493–503, 1974.

Patrick J Heagerty and Yingye Zheng. Survival model predictive accuracy and roc curves. *Biometrics*, 61(1):92–105, 2005.

Graeme L Hickey, Pete Philipson, Andrea Jorgensen, and Ruwanthi Kolamunnage-Dona. Joint modelling of time-to-event and multivariate longitudinal outcomes: recent developments and issues. *BMC medical research methodology*, 16(1):117, 2016.

Graeme L Hickey, Pete Philipson, Andrea Jorgensen, and Ruwanthi Kolamunnage-Dona. joinerml: a joint model and software package for time-to-event and multivariate longitudinal outcomes. *BMC medical research methodology*, 18(1):50, 2018.

Miran A Jaffa, Mulugeta Gebregziabher, and Ayad A Jaffa. A joint modeling approach for right censored high dimensional multivariate longitudinal data. *Journal of biometrics & biostatistics*, 5(4), 2014.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

John P Klein. Semiparametric estimation of random effects using the cox model based on the em algorithm. *Biometrics*, pages 795–806, 1992.

John P Klein and Melvin L Moeschberger. *Survival analysis: techniques for censored and truncated data*. Springer Science & Business Media, 2005.

Haiqun Lin, Charles E McCulloch, and Susan T Mayne. Maximum likelihood estimation in the joint analysis of time-to-event and multiple longitudinal variables. *Statistics in Medicine*, 21(16):2369–2382, 2002a.

Haiqun Lin, Bruce W Turnbull, Charles E McCulloch, and Elizabeth H Slate. Latent class models for joint analysis of longitudinal biomarker and event process data: application to longitudinal prostate-specific antigen readings and prostate cancer. *Journal of the American Statistical Association*, 97(457):53–65, 2002b.

Jean Jacques Moreau. Fonctions convexes duales et points proximaux dans un espace hilbertien. *Comptes rendus hebdomadaires des séances de l'Académie des sciences*, 255: 2897–2899, 1962.

B. N. Mukherjee and S. S. Maiti. On some properties of positive definite toeplitz matrices and their possible applications. *Linear algebra and its applications*, 102:211–240, 1988.

Paul A Murtaugh, E Rolland Dickson, Gooitzen M Van Dam, Michael Malinchoc, Patricia M Grambsch, Alice L Langworthy, and Chris H Gips. Primary biliary cirrhosis: prediction of short-term survival based on repeated patient visits. *Hepatology*, 20(1): 126–134, 1994.

Cécile Proust-Lima, Mbéry Séne, Jeremy MG Taylor, and Hélène Jacqmin-Gadda. Joint latent class models for longitudinal and time-to-event data: A review. *Statistical methods in medical research*, 23(1):74–90, 2014.

Cécile Proust-Lima, Viviane Philipps, and Benoit Liquet. Estimation of extended mixed models using latent classes and latent processes: The r package lcmm. *Journal of Statistical Software, Articles*, 78(2):1–56, 2017. ISSN 1548-7660. doi: 10.18637/jss.v078.i02. URL https://www.jstatsoft.org/v078/i02.

Dimitris Rizopoulos. The r package jmbayes for fitting joint models for longitudinal and time-to-event data using mcmc. *Journal of Statistical Software, Articles*, 72(7):1–46, 2016. ISSN 1548-7660. doi: 10.18637/jss.v072.i07. URL https://www.jstatsoft.org/v072/i07.

Dimitris Rizopoulos and Pulak Ghosh. A bayesian semiparametric multivariate joint model for multiple longitudinal outcomes and a time-to-event. *Statistics in medicine*, 30(12): 1366–1380, 2011.

R. T. Rockafellar. *Conjugate duality and optimization*. Regional conference series in applied mathematics. Society for Industrial and Applied Mathematics, Philadelphia, 1974. ISBN 0-89871-013-8. URL http://opac.inria.fr/record=b1083670. Essentially to be regarded as supplement to the book Convex analysis.

Michael Schemper and Robin Henderson. Predictive accuracy and explained variation in cox regression. *Biometrics*, 56(1):249–255, 2000.

Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of computational and graphical statistics*, 22(2):231–245, 2013.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

Glenn T Sueyoshi. Semiparametric proportional hazards estimation of competing risks models with time-varying covariates. *Journal of econometrics*, 51(1-2):25–58, 1992.

Terry M Therneau and Patricia M Grambsch. The cox model. In *Modeling survival data: extending the Cox model*, pages 39–77. Springer, 2000.

Ryan Tibshirani. Proximal gradient descent and acceleration. *Lecture Notes*, 2010.

Anastasios A Tsiatis and Marie Davidian. Joint modeling of longitudinal and time-to-event data: an overview. *Statistica Sinica*, pages 809–834, 2004.

Hajime Uno, Tianxi Cai, Michael J Pencina, Ralph B D'Agostino, and LJ Wei. On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. *Statistics in medicine*, 30(10):1105–1117, 2011.

Jeroen K Vermunt and Jay Magidson. Latent class models for classification. *Computational Statistics & Data Analysis*, 41(3-4):531–537, 2003.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

Ping Wang, Wei Shen, and Mark Ernest Boye. Joint modeling of longitudinal outcomes and survival using latent growth modeling approach in a mesothelioma trial. *Health Services and Outcomes Research Methodology*, 12(2-3):182–199, 2012.

CF Jeff Wu. On the convergence properties of the em algorithm. *The Annals of Statistics*, 11:95–103, 1983.

Michael S Wulfsohn and Anastasios A Tsiatis. A joint model for survival and longitudinal data measured with error. *Biometrics*, pages 330–339, 1997.

Menggang Yu, Ngayee J Law, Jeremy MG Taylor, and Howard M Sandler. Joint longitudinal-survival-cure models and their application to prostate cancer. *Statistica Sinica*, pages 835–862, 2004.

Lei Yuan, Jun Liu, and Jieping Ye. Efficient methods for overlapping group lasso. *Advances in neural information processing systems*, 24:352–360, 2011.

Zhongheng Zhang, Jaakko Reinikainen, Kazeem Adedayo Adeleke, Marcel E Pieterse, and Catharina GM Groothuis-Oudshoorn. Time-varying covariates and coefficients in cox regression models. *Annals of translational medicine*, 6(7), 2018.

Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–560, 1997.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.