Hindawi
Computational and Mathematical Methods in Medicine
Volume 2021, Article ID 4384035, 11 pages
https://doi.org/10.1155/2021/4384035

Research Article

Joint Modelling Approaches to Survival Analysis via Likelihood-Based Boosting Techniques

Colin Griesbach,[1] **Andreas Groll**,[2] **and Elisabeth Bergherr**[1]

[1]*Chair of Spatial Data Science and Statistical Learning, Georg August University, Germany*
[2]*Department of Statistics, TU Dortmund University, Germany*

Correspondence should be addressed to Colin Griesbach; colin.griesbach@fau.de

Received 30 July 2021; Accepted 15 October 2021; Published 15 November 2021

Academic Editor: Konstantin G. Arbeev

Joint models are a powerful class of statistical models which apply to any data where event times are recorded alongside a longitudinal outcome by connecting longitudinal and time-to-event data within a joint likelihood allowing for quantification of the association between the two outcomes without possible bias. In order to make joint models feasible for regularization and variable selection, a statistical boosting algorithm has been proposed, which fits joint models using component-wise gradient boosting techniques. However, these methods have well-known limitations, i.e., they provide no balanced updating procedure for random effects in longitudinal analysis and tend to return biased effect estimation for time-dependent covariates in survival analysis. In this manuscript, we adapt likelihood-based boosting techniques to the framework of joint models and propose a novel algorithm in order to improve inference where gradient boosting has said limitations. The algorithm represents a novel boosting approach allowing for time-dependent covariates in survival analysis and in addition offers variable selection for joint models, which is evaluated via simulations and real world application modelling CD4 cell counts of patients infected with human immunodeficiency virus (HIV). Overall, the method stands out with respect to variable selection properties and represents an accessible way to boosting for time-dependent covariates in survival analysis, which lays a foundation for all kinds of possible extensions.

1. Introduction

First suggested by [1], joint models were established as a valuable tool for analysing data where event times are measured alongside a longitudinal outcome. One naive approach of evaluating such frequently occurring data structures would be separate modelling, i.e., fitting appropriate models independently for longitudinal and time-to-event data. However, separate modelling neither corrects for event-dependent dropout in longitudinal analysis nor quantifies the relation between a time-dependent covariate and the risk for an event in survival analysis [2]. Various approaches have been proposed to address these issues, one being the Andersen-Gill model [3] for time-varying covariates in survival analysis or two-stage approaches, where longitudinal model fits are included as fixed covariates in time-to-event regression. It has been shown, however, that these methods tend to produce biased results [4, 5]. One solution therefore is combining both the survival and longitudinal models within one single joint likelihood. A wide introduction to this joint modelling framework is presented in [4] including the JM package [6]. Moreover, an evolution of joint model progression up to the year 2004 is provided in [7], and in addition, several Bayesian approaches have been carried out [8–10].

Current joint modelling estimation methods, however, lack clear concepts for proper variable selection and good performance regarding prediction. Moreover they are not feasible for high-dimensional data, in particular where the number of covariates exceeds the number of observations, i.e., $p > n$ problems. In order to overcome these hindrances, an algorithm was initially proposed, where joint models are fitted with gradient boosting techniques, which are known for addressing exactly these issues [11]. Evolved from machine learning as an approach to classification problems

(i) **Initialize** starting values $\widehat{\beta}_0^{[0]}$, $\widehat{\beta}_t^{[0]}$, $\widehat{\beta}_1^{[0]}$, $\widehat{\lambda}^{[0]}$, $\widehat{\beta}_s^{[0]}$, $\widehat{\alpha}^{[0]}$, $\widehat{\gamma}_0^{[0]}$, and $\widehat{\gamma}_t^{[0]}$ with variance-covariance-components $\widehat{\sigma}^{2[0]}$, $\widehat{\mathbf{Q}}^{[0]}$ according to Section 3.3. Choose $(m_{\max,l}, m_{\max,s})$ and execute the following for every possible tuple (m_l, m_s) with $m_l < m_{\max,l}$, $m_s < m_{\max,s}$:

Longitudinal part

(ii) **for** $m = 1$ to m_l **do**

step1: Update fixed effects

For $r = 1, \cdots, p_l$ define $\tilde{\beta}_r := (\widehat{\beta}_0^{[m-1]}, \widehat{\beta}_t^{[m-1]}, \widehat{\beta}_r^{[m-1]})^T$ with $\widehat{\beta}_r^{[m-1]}$ denoting the rth component of $\widehat{\beta}_1^{[m-1]}$. Compute the score vector and Fisher matrix

$$\mathbf{s}_r(\tilde{\beta}_r) = \partial \ell^{\text{pen}} / \partial \tilde{\boldsymbol{\beta}}_r, \mathbf{F}_r(\tilde{\beta}_r) = -\mathbb{E}[\partial^2 \ell^{\text{pen}} / \partial \tilde{\boldsymbol{\beta}}_r \partial \tilde{\boldsymbol{\beta}}_r^T],$$

with respect to the current intercept $\widehat{\beta}_0^{[m-1]}$, time effect $\widehat{\beta}_t^{[m-1]}$, and the rth linear effect $\widehat{\beta}_r^{[m-1]}$. Obtain p_l possible updates

$$\mathbf{u}_r = \mathbf{F}_r(\tilde{\boldsymbol{\beta}}_r)^{-1} \mathbf{s}_r(\tilde{\beta}_r),$$

and find the best performing component $* \in \{1, \cdots, p_l\}$ according to Section 2.2, yielding the update $\mathbf{u}_* = (u_0, u_t, u_*)^T$ containing the update for the effect $*$ with corresponding updates u_0 for intercept and u_t for the time effect. Receive $\widehat{\beta}_0^{[m]}$, $\widehat{\beta}_t^{[m]}$, and $\widehat{\beta}_1^{[m]}$ by updating

$$\widehat{\beta}_0^{[m]} = \widehat{\beta}_0^{[m-1]} + \nu_l u_0, \widehat{\beta}_t^{[m]} = \widehat{\beta}_t^{[m-1]} + \nu_l u_t,$$

$$\widehat{\beta}_r^{[m]} = \begin{cases} \widehat{\beta}_r^{[m-1]} & \text{if } r \neq *, \\ \widehat{\beta}_r^{[m-1]} + \nu_l u_* & \text{if } r = *, \end{cases} \quad r = 1, \cdots, p_l.$$

step2: Update random effects

Receive updates

$$\widehat{\gamma}_0^{[m-1]} \longrightarrow \widehat{\gamma}_0^{[m]}, \widehat{\gamma}_t^{[m-1]} \longrightarrow \widehat{\gamma}_t^{[m]},$$

for random intercepts and slopes in an additional Fisher scoring step on the penalized log-likelihood ℓ^{pen}.

step3: Update variance-covariance components

Update variance-covariance components

$$\widehat{\sigma}^{2[m-1]} \longrightarrow \widehat{\sigma}^{2[m]}, \widehat{\mathbf{Q}}^{[m-1]} \longrightarrow \widehat{\mathbf{Q}}^{[m]}$$

following the description in Section 3.3.

end for

Proceed with estimates $\widehat{\beta}_0^{[m_l]}$, $\widehat{\beta}_t^{[m_l]}$, $\widehat{\beta}_1^{[m_l]}$, $\widehat{\gamma}_0^{[m_l]}$, $\widehat{\gamma}_t^{[m_l]}$ as fixed values.

Survival part

for $m = 1$ to m_s **do**

Update survival effects

For $r = 1, \cdots, p_s + 1$ define $\tilde{\beta}_r := (\widehat{\lambda}^{[m-1]T}, \widehat{\beta}_r^{[m-1]})^T$ with $\widehat{\beta}_r^{[m-1]}$ denoting the rth component of $\widehat{\beta}_s^{[m-1]}$ and $\widehat{\beta}_{p_s+1}^{[m-1]} = \widehat{\alpha}^{[m-1]}$. Compute the score vector and Fisher matrix

$$\mathbf{s}_r(\tilde{\beta}_r) = \partial \ell^{\text{pen}} / \partial \tilde{\boldsymbol{\beta}}_r, \mathbf{F}_r(\tilde{\beta}_r) = -\mathbb{E}[\partial^2 \ell^{\text{pen}} / \partial \tilde{\boldsymbol{\beta}}_r \partial \tilde{\boldsymbol{\beta}}_r^T],$$

with respect to the current baseline hazard $\widehat{\lambda}^{[m-1]}$ and the rth linear effect $\beta_r^{[m-1]}$ or $\widehat{\alpha}^{[m-1]}$, respectively. Obtain $p_s + 1$ possible updates

$$\mathbf{u}_r = \mathbf{F}_r(\tilde{\beta}_r)^{-1} \mathbf{s}_r(\tilde{\beta}_r),$$

and find the best performing effect $* \in \{1, \cdots, p_s + 1\}$ according to Section 2.2, yielding the update $\mathbf{u}_* = (\mathbf{u}_\lambda^T, u_*)^T$ containing the update for the effect $*$ with corresponding baseline hazard update $\mathbf{u}_\lambda$. Receive $\widehat{\lambda}^{[m]}$, $\widehat{\beta}_s^{[m]}$, and $\widehat{\alpha}^{[m]}$ by updating

$$\widehat{\lambda}^{[m]} = \widehat{\lambda}^{[m-1]} + \nu_s \mathbf{u}_\lambda, \widehat{\alpha}^{[m]} = \begin{cases} \widehat{\alpha}^{[m-1]}, & \text{if } * \neq p_s + 1, \\ \widehat{\alpha}^{[m-1]} + \nu_s u_*, & \text{if } * = p_s + 1, \end{cases}$$

$$\widehat{\beta}_r^{[m]} = \begin{cases} \widehat{\beta}_r^{[m-1]} & \text{if } r \neq *, \\ \widehat{\beta}_r^{[m-1]} + \nu_l u_* & \text{if } r = *, \end{cases} \quad r = 1, \cdots, p_s.$$

end for

(iii) **Determine** the best performing tuple $(m_{*,l}, m_{*,s})$ with respect to prediction based on the unpenalized joint log-likelihood ℓ as explained in more detail in Section 3.3.

Algorithm 1: lbbJM.

originally proposed in [12], gradient boosting deals with high-dimensional data and the component-wise updating scheme offers implicit variable selection. The boosting algorithm for joint models was extended [13], but when wanting to lay more focus on the survival side of the model, gradient boosting proved to struggle with time-varying covariates in time-to-event analysis. This has also been observed for pure survival models in [14].

Hence, this work focuses on likelihood-based boosting. First introduced in [15], likelihood-based boosting is

Table 1: Shrinkage and variable selection properties regarding longitudinal and survival outcomes averaged over 100 simulation runs of the low-dimensional scenario.

	β_t (sd)	β_{l1} (sd)	β_{l2} (sd)	β_{l3} (sd)	TP	FDR	m_l^*
True	2	1	2	1			
JM	1.998	0.994	2.008	1.002	—	—	—
	(0.03)	(0.07)	(0.07)	(0.07)			
lbbJM[a]	1.760	0.914	1.922	0.923	1.00	0.23	108.25
	(0.08)	(0.07)	(0.07)	(0.07)			
lbbJM[b]	1.992	0.994	2.008	1.002	—	—	—
	(0.03)	(0.07)	(0.07)	(0.07)			
	α (sd)	β_{s1} (sd)	β_{s2} (sd)	β_{s3} (sd)	TP	FDR	m_s^*
True	0.5	1	2	-2			
JM	0.457	0.903	1.807	-1.800	—	—	—
	(0.04)	(0.08)	(0.12)	(0.12)			
lbbJM[a]	0.390	0.728	1.521	-1.516	1.00	0.27	209.2
	(0.03)	(0.06)	(0.07)	(0.07)			
lbbJM[b]	0.373	0.713	1.500	-1.495	1.00	0.22	196.9
	(0.03)	(0.06)	(0.07)	(0.08)			
glmnet	0.427	0.909	1.833	-1.823	1.00	0.51	—
	(0.03)	(0.07)	(0.11)	(0.10)			

designed to directly maximize a given likelihood where concepts, which gradient boosting implicitly offers, are reproduced artificially for regular optimization methods like Newton algorithms or Fisher scoring. The method was further developed for flexible semiparametric mixed models [16] and for several classes of generalised mixed models [17–20]. The R package GMMBoost [21] covers most of these approaches. Likelihood-based boosting has also been proved useful for survival analysis with time-varying effects [22], and a general overview is given in [23]. Since the random structure plays an important role as a connector between longitudinal and time-to-event data, we additionally incorporate a novel correction step within the estimation procedure for the random effects, which was first suggested in [24, 25] and reduces possible bias arising from wrongly identified random effects.

The contribution of this work is the novel lbbJM boosting algorithm for joint models, which offers the first boosting-based regularization approach for time-dependent covariates in survival analysis and in addition new variable selection mechanics for joint models with focus on time-to-event analysis.

The remainder of this manuscript is structured as follows: Section 2 highlights the overall concepts of both joint modelling and likelihood-based boosting to give a sufficient understanding of the methods used in the following parts. Section 3 then contains a detailed description of the considered joint model together with the proposed boosting algorithm and its computational details. Sections 4 and 5 deal with applying the algorithm to different setups of simulated data as well as to the AIDS dataset [26] included in the JM package. Results and possible extensions are discussed in the final section.

2. Backgrounds

Before the algorithm is presented and discussed in detail, we briefly highlight the concepts of both joint modelling and likelihood-based boosting.

2.1. Joint Models. In general, a joint model consists of two parts, one longitudinal and one survival submodel. A popular view on joint modelling is to choose one model as the *main model*, whereas the other model then features the analysis occurring in the main model. With the primary outcome being longitudinal data, a survival model can be used to correct for event-dependent dropout in longitudinal analysis. For time-to-event data as outcome of interest, additional longitudinal modelling reduces measurement error on the one hand and, on the other hand, extrapolates only on single time points observed longitudinal data to continuous functions which are then included in survival analysis. We will from now on focus on joint models with time-to-event data as primary outcome.

The longitudinal submodel takes the form:

$$y = \eta_l(t, \mathbf{x}) + \varepsilon, \tag{1}$$

where longitudinal outcome y is described by the longitudinal predictor function η_l depending on time t and a set of covariates $\mathbf{x}$. Although t can be included in $\mathbf{x}$, we will highlight it in the context of joint models, as the role of t is of greater importance. In the survival submodel, the hazard

$$\lambda(t \mid \mathbf{x}) = \lambda_0(t) \exp\left(\eta_s(\mathbf{x}) + \alpha\eta_l(t, \mathbf{x})\right), \tag{2}$$

is modelled by a baseline hazard $\lambda_0(t)$ with multiplicative effects described by the survival predictor function η_s. In addition, the longitudinal predictor η_l is reappearing in the survival model, this time scaled by a factor α. The parameter α thus quantifies the association of the two submodels and is therefore called the *association parameter*. It can be interpreted as the impact a time-varying longitudinal covariate has on the hazard for an event.

Parameter estimation for such joint models can be done in various ways. Two common methods are two-stage and joint likelihood approaches, respectively. In the former, the longitudinal model is estimated with the estimation method of choice leading to the model fit $\widehat{\eta}_l$, which is then carried forward as fixed covariate into survival analysis. In the latter, longitudinal and survival submodels are combined in a single joint likelihood. Let $i = 1, \cdots, n$ denote clusters and $j = 1, \cdots n_i$ the repeated measurements. Assuming independent data generating processes for both submodels, the joint likelihood can then be written as

$$L(y, T, \delta) = \prod_{i=1}^{n} \left(\prod_{j=1}^{n_i} f_l\left(y_{ij} \mid \eta_l\right) \right) f_s(T_i, \delta_i \mid \alpha, \eta_l, \eta_s, \lambda_0), \tag{3}$$

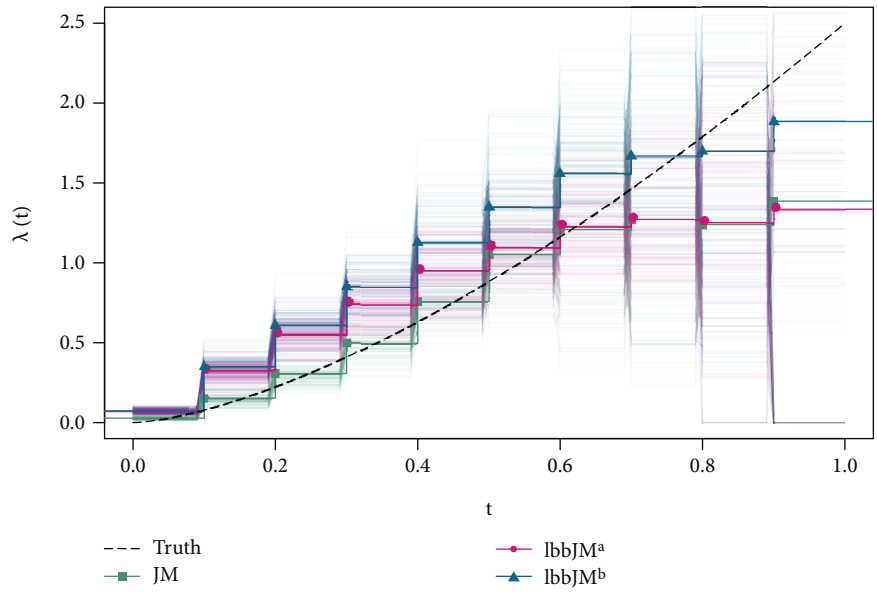

Figure 1: Piecewise-constant baseline hazard estimates with $K = 10$ by JM, lbbJM^a and lbbJM^b averaged over 100 simulation runs of the low dimensional scenario.

Table 2: Shrinkage and variable selection properties regarding the longitudinal and survival outcomes averaged over 100 simulation runs of the high-dimensional scenario.

	β_t (sd)	β_{l1} (sd)	β_{l2} (sd)	β_{l3} (sd)	TP	FDR	m_l^*
True	2	1	2	1			
lbbJM^a	1.748	0.868	1.843	0.875	1.00	0.36	124.2
	(0.20)	(0.14)	(0.14)	(0.14)			
lbbJM^b	1.991	1.008	1.982	1.011	—	—	—
	(0.08)	(0.13)	(0.16)	(0.15)			
	α (sd)	β_{s1} (sd)	β_{s2} (sd)	β_{s3} (sd)	TP	FDR	m_s^*
True	0.5	1	2	-2			
lbbJM^a	0.307	0.512	1.242	-1.216	1.00	0.70	136.7
	(0.06)	(0.12)	(0.16)	(0.13)			
lbbJM^b	0.285	0.498	1.215	-1.191	1.00	0.67	127.0
	(0.05)	(0.13)	(0.15)	(0.13)			
glmnet	0.293	0.627	1.449	-1.422	1.00	0.83	—
	(0.08)	(0.18)	(0.30)	(0.28)			

with densities f_l and f_s for the longitudinal and survival submodels and time-to-event outcome $(T, \delta) = (T_i, \delta_i)_{i \in \mathbb{N}}$. Regular inference is now done by maximizing (3) using appropriate maximization methods, the most prominent one being EM algorithms.

2.2. Likelihood-Based Boosting. We intend to give a short description of the underlying mechanics used in the following section. The overall concept of likelihood-based boosting is to create an iterative and component-wise updating scheme, which eventually converges to a maximum likelihood estimator but is stopped early in order to prevent overfitting. Let β model be the effect of p covariates. Likelihood-based boosting maximizes a given log-likelihood $l(\beta)$ by component-wise Fisher scoring in the following way:

For each covariate $r \in \{1, \cdots, p\}$ consider the subvector β_r containing only the coefficients referring to the rth covariate. We compute the score vector and Fisher matrix as

$$\mathbf{s}_r(\beta_r) = \frac{\partial l(\beta)}{\partial \beta_r}, \quad \mathbf{F}_r(\beta_r) = -\mathbb{E}\left[\frac{\partial^2 l(\beta)}{\partial \beta_r \partial \beta_r^T}\right], \tag{4}$$

and obtain a possible update

$$\mathbf{u}_r := \mathbf{F}_r(\beta_r)^{-1} \mathbf{s}_r(\beta_r), \tag{5}$$

for the rth component. Now, we determine the best performing covariate with respect to likelihood maximization, i.e., find the component

$$* = \underset{r=1,\cdots,p}{\arg\max}\, l\left(\tilde{\beta}_r\right), \quad \tilde{\beta}_r = \left(\beta_1, \cdots, \beta_r + \mathbf{u}_r, \cdots, \beta_p\right) \tag{6}$$

where the corresponding update yields the biggest improvement of the likelihood. One receives a new model fit by weakly updating this best performing component, i.e., by scaling with a factor ν, the so called *step length*:

$$\beta_r^{\text{new}} = \begin{cases} \beta_r & \text{if } r \neq *, \\ \beta_r + \nu \mathbf{u}_r & \text{if } r = *, 0 < \nu \leq 1, \end{cases} \quad r = 1, \cdots, p. \tag{7}$$

The step length ν is controlling the weakness of the update to prevent overfitting and give every covariate a chance for selection. A popular choice in the literature is $\nu = 0.1$.

Table 3: Averaged computation times for one single model fit (in seconds).

Setup	JM	glmnet	lbbJM[a]	lbbJM[b]
Low	110.00	149.15	15776.16	43.76
High	—	156.44	4072.80	248.08

Table 4: Structure of the dataset with primary outcomes for the joint analysis in the three columns on the left.

y	T	δ	t	Drug	Gender	AZT	prevOI	ID
10.67	16.97	0	0	ddC	Male	Intolerance	AIDS	1
8.43	16.97	0	6	ddC	Male	Intolerance	AIDS	1
9.43	16.97	0	12	ddC	Male	Intolerance	AIDS	1
6.32	19.00	0	0	ddI	Male	Intolerance	noAIDS	2
8.12	19.00	0	6	ddI	Male	Intolerance	noAIDS	2
4.58	19.00	0	12	ddI	Male	Intolerance	noAIDS	2
5.00	19.00	0	18	ddI	Male	Intolerance	noAIDS	2
3.46	18.53	0	0	ddI	Female	Intolerance	AIDS	3
3.61	18.53	0	2	ddI	Female	Intolerance	AIDS	3
6.16	18.53	1	6	ddI	Female	Intolerance	AIDS	3
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Repeating this updating process for a sufficiently large number of iterations leads to the regular maximum likelihood estimator for β. But instead the algorithm is stopped early to gain better prediction performance and variable selection. The optimal amount of iterations actually is a tuning parameter of the method and can be determined via cross-validation or by focusing on information criteria like AIC or BIC [27, 28].

3. Boosting Joint Models

3.1. The Model. Before we introduce the boosting algorithm, we describe the specific joint model. With $i = 1, \cdots, n$ denoting the individual and $j = 1, \cdots, n_i$ a single specific measurement, the longitudinal submodel is given by

$$y_{ij} = \eta_l(t_{ij}, \mathbf{x}_{li}) + \varepsilon_{ij} = \beta_0 + \beta_t t_{ij} + \beta_l^T \mathbf{x}_{li} + \gamma_{0i} + \gamma_{ti} t_{ij} + \varepsilon_{ij}, \tag{8}$$

where y_{ij} is modelled by η_l depending on specific measurement times t_{ij} and longitudinal time-independent covariates $\mathbf{x}_{li} \in \mathbb{R}^{p_l}$. This represents a standard linear mixed model with intercept β_0 and fixed linear effects β_t and β_l of time and baseline covariates as well as individual specific random effects γ_{0i} and γ_{ti} with $(\gamma_{0i}, \gamma_{ti}) \sim \mathcal{N}^{\otimes 2}(\mathbf{0}, \mathbf{Q})$. The error terms ε_{ij} are assumed to follow a normal distribution with $\mathbb{E}[\varepsilon_{ij}] = 0$ and $\mathrm{Var}(\varepsilon_{ij}) = \sigma^2 > 0$.

Please note that the model can be additionally extended to interaction effects of time t_{ij} with baseline covariates $\mathbf{x}_{ti} \in \mathbb{R}^{p_t}$ by including the term $\boldsymbol{\beta}_t^T \mathbf{x}_{ti} t_{ij}$ in (8). This results in slightly adjusted integrals in the survival part and is omitted in the following for the sake of better readability.

In the survival part, the individual hazard

$$\lambda_i(t) = \lambda_0(t) \exp\left(\eta_s(\mathbf{x}_{si}) + \alpha \eta_l(t, \mathbf{x}_{li})\right), \tag{9}$$

is modelled with the survival predictor $\eta_s(\mathbf{x}_{si}) = \beta_s^T \mathbf{x}_{si}$ containing additional linear effects β_s of baseline covariates $\mathbf{x}_{si} \in \mathbb{R}^{p_s}$. To execute a full likelihood approach, the baseline hazard

$$\lambda_0(t) = \sum_{k=1}^{K} \lambda_k \mathbf{1}_{I_k}(t), \tag{10}$$

is chosen to be piecewise-constant depending on the number of segments K and their exact locations $I_k = [t_{k-1}, t_k)$ with $t_0 = 0$ and $t_k = \max(\mathbf{T})$ for $k = 1, \cdots, K$. The collection of values for the baseline hazard is denoted in $\lambda = (\lambda_k)_{k=1,\cdots,K}$. Later, we will choose K between 7 and 10 in order to guarantee substantially more flexibility than a constant baseline hazard without becoming computationally too demanding.

Given two formulas (8) and (9), we can now calculate the joint log-likelihood. Let $\mathbf{y} = (y_{ij})_{i=1,\cdots,n, j=1,\cdots,n_i}$ denote the collection of all longitudinal measurements. Assuming the time-to-event process is conditionally independent from the longitudinal random structure, the joint likelihood can be decomposed into a longitudinal and a survival term. Set $\boldsymbol{\gamma}^T = (\boldsymbol{\gamma}_i)_{i=1,\cdots,n}$ with $\boldsymbol{\gamma}_i^T = (\gamma_{0i}, \gamma_{ti})$ and $\vartheta_l := (\beta_0, \boldsymbol{\beta}_l^T, \beta_t, \boldsymbol{\gamma}^T)^T$. Furthermore, τ contains information on variance-covariance components σ^2 and $\mathbf{Q}$. The unpenalized longitudinal log-likelihood is

$$\ell_l(\vartheta_l, \sigma^2 \mid \mathbf{y}) = \sum_{i=1}^{n} \sum_{j=1}^{n_i} \log \phi\left(y_{ij} \mid \eta_l(t_{ij}, \mathbf{x}_{li}), \sigma^2\right), \tag{11}$$

where $\phi(\cdot \mid m, v)$ denotes the density of a normal distribution with mean m and variance v. Laplace approximation follows [29] and then leads to an additional quadratic penalty term for the random effects yielding the penalized log-likelihood:

$$\ell_l^{\mathrm{pen}}(\vartheta_l, \tau \mid \mathbf{y}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{n_i} \log \phi\left(y_{ij} \mid \eta_l(t_{ij}, \mathbf{x}_{li}), \sigma^2\right) - \frac{1}{2} \gamma_i^T \mathbf{Q} \gamma_i \right). \tag{12}$$

Note that for the penalized log-likelihood τ substitutes σ^2 as an argument, since the penalized log-likelihood additionally contains information of the variance matrix $\mathbf{Q}$.

On the other hand, for given survival outcome $(\mathbf{T}, \delta) = (T_i, \delta_i)_{i=1,\cdots,n}$ with event times T_i and censoring indicator δ_i, the survival log-likelihood takes the well-known form:

$$\ell_s(\vartheta_s \mid \mathbf{T}, \boldsymbol{\delta}) = \sum_{i=1}^{n} \delta_i \log \lambda_i(T_i \mid \eta_l, \eta_s, \alpha, \lambda) - \int_0^{T_i} \exp\left(\lambda_i(t \mid \eta_l, \eta_s, \alpha, \lambda)\right) dt, \tag{13}$$

Table 5: Shrinkage and variable selection properties by the different packages for model (22).

	β_0	β_l	β_{t1}	β_{t2}	α	β_{s1}	β_{s2}	β_{s3}
JM	6.97	0.49	−0.18	<0.01	−0.24	0.31	0.09	0.66
lbbJM[a]	6.95	0.26	−0.05	0	−0.13	0	0	0.73
lbbJM[b]	6.95	0.48	−0.16	−0.02	−0.18	0.03	0	0.61
glmnet	—	—	—	—	−0.15	0.31	0.09	0.81
p value (JM)	<0.01	0.26	<0.01	0.98	<0.01	0.23	0.61	<0.01

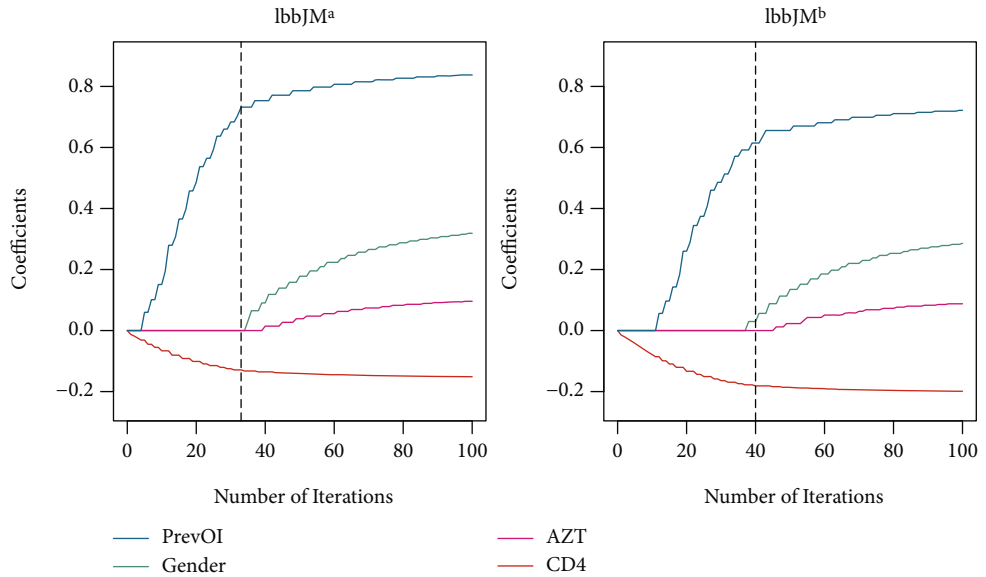

Figure 2: Coefficient progression in the survival part for lbbJM[a] ((a), with $m_{*,l} = 10$) and lbbJM[b] (b).

with $\vartheta_s := (\lambda^T, \alpha, \beta_s^T)^T$. For $\vartheta := (\vartheta_l^T, \vartheta_s^T)^T$, we finally receive the penalized and unpenalized joint log-likelihood:

$$\begin{aligned} \ell(\vartheta, \sigma^2 \mid \mathbf{y}, \mathbf{T}, \delta) &= \ell_l(\vartheta_l, \sigma^2 \mid \mathbf{y}) + \ell_s(\vartheta_s \mid \mathbf{T}, \delta), \\ \ell^{\text{pen}}(\vartheta, \tau \mid \mathbf{y}, \mathbf{T}, \delta) &= \ell_l^{\text{pen}}(\vartheta_l, \tau \mid \mathbf{y}) + \ell_s(\vartheta_s \mid \mathbf{T}, \delta). \end{aligned} \tag{14}$$

3.2. The Algorithm. The following lbbJM algorithm (likelihood-based boosting for joint models) describes a way to fit the formulated joint model by likelihood-based boosting methods discussed in Section 2.2.

3.3. Computational Details of the Algorithm. In general, the algorithm carries out a hybrid between a two-stage and a joint likelihood approach. For one single tuple (m_l, m_s), the fitting procedure goes as follows: In a first step, the longitudinal submodel is boosted up to m_l iterations using the lbbLMM boosting algorithm [25]. The received estimates are carried forward into the survival model, where another boosting process up to m_s iterations takes place. This fitting process is carried out for any tuples (m_l, m_s) with $m_l < m_{\max,l}$, $m_s < m_{\max,s}$, where $m_{\max,l}$ and $m_{\max,s}$ are prespecified maximum numbers of iterations per submodel. For every of these possible combinations of stopping iterations, the corresponding estimates are evaluated based on the joint likelihood using test data, which can be achieved via cross-validation or bootstrapping. Hence, the algorithm uses two-stage fitting but joint likelihood evaluation. We give a detailed description for both parts. Exact formulas for all appearing variants of score vectors and Fisher matrices can be found in the supplementary material. Please note that due to the component-wise updating scheme in both submodels, the lbbJM algorithm works with arbitrarily high numbers of candidate variables and is therefore not confined to low-dimensional data structures.

For starting values, the parameters, which actually underlie the boosting process, are necessarily set to zero, thus $\widehat{\beta}_l^{[0]} = \widehat{\beta}_s^{[0]} = \widehat{\alpha}^{[0]} = \mathbf{0}$. The baseline hazard is initialized with the intercept estimator $\widehat{\lambda}^{[0]} = (\sum_i \delta_i / \sum_i T_i)_{k=1,\cdots,K}$. The remaining values are extracted from a standard linear mixed model for time and random effects

$$y_{ij} = \beta_0 + \beta_t t_{ij} + \gamma_{0i} + \gamma_{ti} t_{ij} + \varepsilon_{ij}, \tag{15}$$

by using, e.g., the function lme from the R package nlme.

For boosting longitudinal fixed effects, for convenience, we omit the iteration index as well as the hat indicating estimated values in the following subsections. In the first step of the longitudinal part, the effects β_l follow the classical component-wise likelihood-based boosting procedure. In

each iteration, an update for every single covariate r together with intercept β_0 and time effect β_t is computed, leading to p_l three-dimensional updates. Selection of the best performing component is then performed either by selecting the component yielding the optimal likelihood maximization or lowest information criteria like AIC or BIC, which minimizes the model complexity rather than residuals. The linear effect β_t is excluded from the selection process, as it holds a very important role in a joint model and should always be included.

For updating random effects, after the best performing fixed effect from β_l was updated, in every iteration, an update for the random effects is executed separately. This means that the score vector and Fisher matrix

$$\mathbf{s}_{\text{ran}}(\gamma) = \frac{\partial \ell^{\text{pen}}}{\partial \gamma}, \quad \mathbf{F}_{\text{ran}}(\gamma) = -\mathbb{E}\left[\frac{\partial^2 \ell^{\text{pen}}}{\partial \gamma \partial \gamma^T}\right], \tag{16}$$

have to be derived in order to execute the update

$$\gamma^{\text{new}} = \gamma + \mathbf{C}\mathbf{F}_{\text{ran}}(\gamma)^{-1}\mathbf{s}_{\text{ran}}(\gamma). \tag{17}$$

The matrix $\mathbf{C}$ is a correction matrix which prevents from potential correlations between the random effects estimates and any observed covariates [25], and its derivation can be traced in more detail in the supplementary material.

For updating variance-covariance components, the covariance matrix $\mathbf{Q}$ of the random effects is updated with an approximate EM algorithm using the posterior curvatures $\mathbf{F}_{ii}$ of the random effects model [30]. Receive an update by computing

$$\mathbf{Q} = \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{F}_{ii}^{-1} + \gamma_i \gamma_i^T\right). \tag{18}$$

The current longitudinal model error is obtained by setting $\text{Var}(\mathbf{y} - \boldsymbol{\eta}_l)$.

For boosting the association parameter and survival effects, once the longitudinal part was updated in up to m_l iterations, the algorithm proceeds to boost the effects β_s and α. Although being of different structures, the association parameter α is boosted alongside the effects β_s, meaning the algorithm decides whether the association or some baseline survival effect is updated, based on which parameter leads to the best likelihood improvement. This means, the linear effect of the whole longitudinal trajectory is also scaled by the step length ν when being updated within the selection step, which minimizes the chance of potential overfitting also for the association parameter. An alternative method would be choosing just from the effects in β_s and updating α in an additional step by optimizing the current likelihood. This approach was used in [11] and treats α as a nuisance parameter, which might not be satisfactory with regards to the importance of α. Again, only the update for α and β_s is scaled by the step length ν_s. The baseline hazard λ receives a full update.

For step lengths, apart from the stopping iterations, the step lengths are tuning parameters of the boosting algorithm. Although there is some effort in focusing on adaptive step lengths recently, we chose to set both step lengths to the constant value $\nu_l = \nu_s = 0.1$. The exact choice of the step length factor is of minor importance as long as it is sufficiently small to ensure proper performance. Setting it to 0.1 is an established choice in the boosting literature [31, 32].

For stopping iterations, since the step lengths are chosen to be constant, the tuple (m_l, m_s) is the main tuning parameter of the boosting algorithm. In regular boosting with only one iteration index, it is convenient to check for every single iteration and take the estimates from the estimation count leading to the best prediction. In the present two-dimensional case, this would mean finding

$$(m_{*,l}, m_{*,s}) = \underset{(m_l,m_s)\in\mathscr{M}}{\arg\max}\, \ell\left(\widehat{\vartheta}^{[m_l,m_s]} \mid \mathscr{X}^{\text{test}}\right), \tag{19}$$

with $\mathscr{M} := \{1, \cdots, m_{\text{max,l}}\} \times \{1, \cdots, m_{\text{max,s}}\}$, $\widehat{\vartheta}^{[m_l,m_s]}$ denoting the vector of estimates received via the tuple (m_l, m_s) of total iterations and $\mathscr{X}^{\text{test}}$ a complete set of test data for evaluation. Problem (19) is then solved via k-fold cross-validation. But since checking for every single tuple $(m_l, m_s) \in \mathscr{M}$ implies a very high computational effort, we suggest to coarsen the grid and check for fewer possible stopping iterations in the longitudinal part, e.g., $m_l \in \{10,20,30, \cdots, m_{\text{max,l}}\}$. Because of the two-stage-approach nature of the algorithm, we still can check for every single $m_s \in \{1, \cdots, m_{\text{max,s}}\}$ without gaining additional computational effort. Furthermore, parallel computing can be executed in order to minimize computational demand.

4. Simulations

We evaluate the lbbJM algorithm with a simulation study. The aim is to assess estimation and shrinkage characteristics in general as well as variable selection properties and performance in high dimensional, i.e., $p > n$ settings. The lbbJM algorithm is included in two variants. While lbbJM$^{\text{a}}$ executes the full approach as depicted in Section 3.2, lbbJM$^{\text{b}}$ performs a shortened two-stage procedure where the longitudinal submodel is fitted in advance using regular maximum likelihood inference and does not underlie any regularization. The exact lbbJM$^{\text{b}}$ algorithm is depicted in detail in the supplementary material. We additionally include the JM package as state of the art for convenient estimation of joint models as well as the glmnet package, which offers elastic net penalization for start-stop-data and therefore an alternative approach for regularization of time-dependent covariates in survival analysis. None of the competitors are completely suitable for a benchmark comparison and are viewed as reference points for the specific objectives addressed by the lbbJM algorithm. Regarding glmnet, as an alternative approach to regularization of time-dependent covariates, shrinkage and variable selection properties are of interest, although it focuses solely on survival analysis. JM in addition offers unregularized effect estimates with corresponding

significance indicator but is neither suitable for high-dimensional setups nor offers variable selection.

4.1. Setup. The simulations are executed according to the model described in Section 3.1 with $n \in \{100,500\}$ and $n_i = 5$ using inversion sampling, which is explained in detail in the supplementary material. The prespecified true parameter values are

$$\beta_0 = 1, \quad \beta_t = 2, \quad \beta_l^T = (2, 1, 2), \quad \beta_s^T = (1, 2, -1), \quad \alpha = 0.5, \tag{20}$$

with variance components

$$\sigma = 0.1, \quad \mathbf{Q} = \begin{pmatrix} 2 & 0.1 \\ 0.1 & 0.3 \end{pmatrix}. \tag{21}$$

The entries of the covariate vectors $\mathbf{x}_{li}$ and $\mathbf{x}_{si}$ are drawn independently from the standard normal distribution $\mathcal{N}(0, 1)$. In addition to the informative covariates with effects β_l and β_s, the covariate vectors $\mathbf{x}_{li}$ and $\mathbf{x}_{si}$ are expanded with noninformative noise variables until the chosen numbers p_l and p_s of total dimensions are reached. These additional noise variables are included to evaluate variable selection properties and robustness of the approach in case of a misspecified model. The baseline hazard is chosen as $\lambda_0(t) = 2.5t^{1.5}$ and given the censoring mechanism described in the supplementary material, the chosen parameter values result in an average censoring rate of ≈50%.

Overall, we consider two scenarios. One low-dimensional setup with $n = 500$ and $p_l = p_s = 9$ mimicking a more common data structure and one high-dimensional setup, where the number of covariates included in the survival submodel exceeds the number of individuals so that conventional approaches like JM fail to return results.

For the computation, JM and lbbJM use $K = 10$ with equidistant knot placement. The grid $\mathcal{M} := \{25, 50, \cdots, 500\} \times \{1, 2, \cdots, 1000\}$ is specified for possible tuples of stopping iterations and the optimal regularization parameter for glmnet is determined by the function cv.glmnet() based on 10-fold cross-validation.

4.2. Results. Since the compared estimation routines follow different approaches targeting various objectives from regular maximum likelihood estimation in joint models to regularization in pure time-to-event analysis, we focus on plain coefficient estimates averaged over 100 independent simulation runs in order to asses estimation characteristics. Variable selection properties are evaluated by considering share of true positives (TP) and false discovery rate (FDR).

For low-dimensional setup ($n = 500$, $p_s = 9$), Table 1 depicts the results for the low-dimensional setup. In the longitudinal submodel, lbbJM[a] has small shrinkage and therefore offers variable selection with a rather low false discovery rate of 0.23 but still selects all informative variables. The time effect β_t receives comparatively high shrinkage, since time, as a cluster-varying variable, adds more information to the model. Overall, the longitudinal submodel is boosted up to 108.25 iterations on average yielding rather strongly shrunk coefficient estimates. Please note that the results for lbbJM[b] are simply obtained by lme() and very similar to JM. In the survival part, both boosting approaches substantially outperform glmnet in terms of variable selection while again receiving also more shrinkage. Due to the comparatively rough baseline hazard depicted in Figure 1, all full likelihood approaches, i.e., JM and lbbJM, receive additional shrinkage which is unaffected by possible regularization. As glmnet uses the partial likelihood, there are no estimates for the baseline hazard function available and the small elastic net penalty of $\lambda^* = 0.003$ on average also results in weaker performance regarding the rate of false positives. Overall, the lbbJM approaches yield satisfactory results regarding both regularization and variable selection. The effect estimates clearly reflect the true values approximately obtained by JM while simultaneously receiving sufficiently large shrinkage for decent performance of identifying influential covariates in both the longitudinal and survival submodels.

For high-dimensional setup ($n = 100$, $p_s = 100$), Table 2 depicts the results for the high-dimensional setup. As expected, estimates in the high-dimensional setup are regularized stronger and therefore experience more shrinkage. Again, the boosting approaches contained in lbbJM yield better variable selection properties and slightly more regularized coefficient estimates, although results seem to align with increasing dimensions. Note that JM is not capable of handling high-dimensional data structures and is therefore not included in the high-dimensional setup at all.

For computational effort, Table 3 shows estimates for elapsed computation time of each routine. Times are measured in seconds and depict the computation time for one single model fit which was carried out on a 2×2.66 GHz-*6-Core Intel Xeon* CPU (*64 GB* RAM). As expected, the full boosting approach executed in lbbJM[a] comes with high computational costs similar to [11]. Note that the runtimes are higher in the low-dimensional scenario as the overall number of clusters is higher ($n = 500$) leading to an increased burden in the already time-consuming longitudinal boosting procedure. The reduced approach lbbJM[b], however, runs considerably faster and is therefore more desirable as long as research focus lies solely on the time-to-event analysis.

5. Application

We showcase the lbbJM algorithm by applying it to the 1994 AIDS data [26]. The study is aimed at comparing the two antiretroviral drugs, didanosine (ddI) and zalcitabine (ddC), based on a collective of 467 patients infected with human immune deficiency virus (HIV) who were either intolerant to or failed a previous treatment with Zidovudine (AZT). Alongside several baseline covariates, the square root CD4 cell count was recorded at study entry and in multiple follow-ups after 2, 6, 12, and 18 months, respectively. The CD4 cells are attacked by the virus and thus decrease over time for infected patients; hence, they are a widely used surrogate for disease progression. In addition to the

longitudinal outcome, 188 patients died during the time of the study leading to 188 observed and 279 censored events. The structure of the data is depicted in Table 4.

We formulate the joint model

$$\mathrm{CD4}_i(t) = \eta_{\mathrm{l}i}(t) + \varepsilon_i(t) = \beta_0 + \beta_{\mathrm{l}}\mathrm{drug}_i + \beta_{t1}t + \beta_{t2}(t \cdot \mathrm{drug}_i) + \gamma_{0i} + \gamma_{ti}t + \varepsilon_i(t), \tag{22}$$

$$\lambda_i(t) = \lambda_0(t)\ \exp\left(\beta_{\mathrm{s}1} \cdot \mathrm{gender}_i + \beta_{\mathrm{s}2} \cdot \mathrm{AZT}_i + \beta_{\mathrm{s}3} \cdot \mathrm{prevOI}_i + \alpha\eta_{\mathrm{l}i}(t)\right). \tag{23}$$

The CD4 cell count $\mathrm{CD4}_i(t)$ for $i = 1, \cdots, 467$ is described by a linear mixed model with random intercepts, random slopes for time t and linear effects of time, drug and an additional interaction between time and drug. The *true*, i.e. modelled by $\eta_{\mathrm{l}}(t)$, underlying profile of the CD4 cell count is then included together with the remaining baseline covariates in the Cox full likelihood model, thus the model error $\varepsilon(t)$ is eliminated. Here, drug is a dummy for ddI, gender for female gender, AZT for failure of Zidovudine therapy, and prevOI for prevalence of AIDS. The number of segments for the baseline hazard was chosen to be $K = 7$.

We fit model (22) with the same methods as already used in the simulation study. The tuning parameters of the boosting algorithm were chosen to be $\nu_{\mathrm{l}} = \nu_{\mathrm{s}} = 0.1$ for step lengths and $\mathscr{M}^a := \{5, 10, \cdots, 100\} \times \{1, 2, \cdots, 250\}$ for the grid of possible tuples of stopping iterations for lbbJM$^{\mathrm{a}}$ and $\mathscr{M}^b := \{1, 2, \cdots, 250\}$ for lbbJM$^{\mathrm{b}}$. Again, glmnet was tuned using the cv.glmnet() function and all regularization approaches are based on 10-fold cross validation.

For lbbJM$^{\mathrm{a}}$, $m_{*,\mathrm{l}} = 10$ and $m_{*,\mathrm{s}} = 33$ formed the best performing tuple of stopping iterations. The two-stage approach lbbJM$^{\mathrm{b}}$ used $m_{*,\mathrm{s}} = 40$ and the optimal penalization parameter for glmnet turned out as $\lambda^*_{\mathrm{pen}} = 0.0014$. The corresponding coefficient estimates are shown in Table 5. Overall, the results reflect what was already observed in the simulation study. While glmnet shows quite conservative shrinkage properties where every variable is included in the final model, the lbbJM approaches tend to stop rather early yielding bigger shrinkage and in addition effects, which did not get selected at all. In order to give some point of reference regarding variable selection properties, the p values computed by JM are included in Table 5. The selected effects align quite nicely with being significant according to JM. While lbbJM$^{\mathrm{a}}$ only includes the variable drug with only half the effect size, lbbJM$^{\mathrm{b}}$ additionally sees a quite tiny impact of female gender.

The corresponding coefficient progressions for the survival submodel are visualized in Figure 2. Both algorithms update the coefficients referring to the longitudinal CD4 cell profile and the variable prevOI right away and therefore see a strong connection between the risk for death and the CD4 cell count as well as whether or not a patient has AIDS. Due to early stopping, lbbJM$^{\mathrm{a}}$ selects neither of the two remaining covariates into the final model. lbbJM$^{\mathrm{b}}$ includes one variable more, gender, which however only has a very small effect.

6. Outlook and Discussion

Overall, the lbbJM algorithm introduces a novel boosting-based regularization scheme to joint models focusing on survival analysis as well as to Cox models with time-dependent covariates. The method fits in well among alternative routines and especially stands out with respect to variable selection properties. Due to its clear advantage regarding computational effort as depicted in Table 3, lbbJM$^{\mathrm{b}}$ is the preferred routine when research interest clearly lies on time-to-event analysis, whereas lbbJM$^{\mathrm{a}}$ is capable of regularizing both submodels simultaneously. Besides the good results regarding variable selection, it is also expected that the proposed boosting methods can improve the predictive power of a joint model, since boosting algorithms are, due to their model tuning based on test errors, primarily used for prediction. A thorough investigation of improving and evaluating prediction performance of a joint model by several boosting techniques remains an interesting task.

Still, the presented foundation is of a comparatively simple nature and possible extensions include more flexible modelling, e.g., based on P-splines which allow smooth effects of time-dependent as well as time-independent covariates and can additionally include time-varying effects as well as possibly time-varying association structures. As the current algorithm is confined to a time-constant association parameter α, an extension to $\alpha(t)$ would increase the flexibility of the model. While it is usually difficult to disentangle a time-dependent association parameter from the longitudinal trajectory for parameter estimation, the lbbJM algorithm could potentially avoid these identification issues due to its clear separation in a longitudinal and a survival boosting process.

Similar regularization-based approaches have been proved useful for time-to-event settings [22, 33], and it can be assumed that the presented method could benefit from these concepts as well.

Moreover, the presented work only lays a foundation to several extensions addressing known issues for both boosting and joint modelling. It represents an accessible way to boosting for time-dependent covariates in survival analysis. Gradient boosting has known limitations in this matter, and although efforts for a framework to overcome these issues are made [34], things rather quickly become technical and the proven robustness of likelihood-based boosting represents a flexible and far more intuitive alternative. Further developments in this direction could include multiple time-dependent covariates based on the two-stage approach of lbbJM$^{\mathrm{b}}$, where additional effort is necessary in order to provide a fair competition between time-dependent and time-independent covariates within the selection procedure.

Furthermore, the component-wise updating process is capable of including allocation mechanisms into the algorithm. While the lbbJM algorithm is due to its two-stage fitting process fairly robust to estimate models, where one candidate variable is assigned to both submodels, this kind of specification is not advised in general as identification issues may arise and a proper interpretation of the resulting model might be challenging. However, it is usually tricky to

decide, whether a covariate should be included in the longitudinal or the survival part of the model and these decisions are often made using prior knowledge. An allocation routine based on likelihood maximization could therefore greatly improve joint model inference and would additionally eliminate the two-stage nature of the lbbJM algorithm. This would not only decrease the computational burden but also provide a far more flexible algorithm allowing for regularization, variable selection, and allocation.

Data Availability

The R code data used to support the findings of this article are included within the supplementary information files.

Conflicts of Interest

The authors declare that there is no conflict of interest.

Acknowledgments

The work on this article was supported by the DFG (Deutsche Forschungsgemeinschaft; Projekt WA 4249/2-1) and the Volkswagen Foundation. We further acknowledge support by the Open Access Publication Funds of the Göttingen University.

Supplementary Materials

Supplementary Materials: a more detailed discussion on the computational details as well as the lbbJM^b routine and the simulation algorithm can be found in the web-appendix. Additionally, the corresponding R code is available in the supplementary files. *(Supplementary Materials)*

References

[1] M. S. Wulfsohn and A. A. Tsiatis, "A joint model for survival and longitudinal data measured with error biometrics," *Biometrics*, vol. 53, p. 330, 1997.

[2] X. Guo and B. P. Carlin, "Separate and joint modeling of longitudinal and event time data using standard computer packages," *The american statistician*, vol. 58, pp. 16–24, 2004.

[3] P. K. Andersen and R. D. Gill, vol. 10, pp. 1100–1120, 1982.

[4] D. Rizopoulos, *Joint Models for Longitudinal and Time-to-Event Data: With Applications in R; 6 of Chapman & Hall/CRC Biostatistics Series*, CRC Press, Boca Raton, 2012.

[5] M. J. Sweeting and S. G. Thompson, "Joint modelling of longitudinal and time-to-event data with application to predicting abdominal aortic aneurysm growth and rupture," *Biometrical Journal*, vol. 53, no. 5, pp. 750–763, 2011.

[6] D. Rizopoulos, "JM: an R package for the joint modelling of longitudinal and time-to-event data," *Journal of Statistical Software*, vol. 35, no. 9, pp. 1–33, 2010.

[7] A. A. Tsiatis and M. Davidian, "Joint modeling of longitudinal and time-to-event data: an overview," *Statistica Sinica*, vol. 14, pp. 809–834, 2004.

[8] M. Köhler, N. Umlauf, A. Beyerlein, C. Winkler, A. G. Ziegler, and S. Greven, "Flexible Bayesian additive joint models with an application to type 1 diabetes research," *Biometrical Journal*, vol. 59, pp. 1144–1165, 2017.

[9] R. Martins, G. L. Silva, and V. Andreozzi, "Joint analysis of longitudinal and survival AIDS data with a spatial fraction of long-term survivors: a Bayesian approach," *Biometrical Journal*, vol. 59, pp. 1166–1183, 2017.

[10] M. Rué, E.-R. Andrinopoulou, D. Alvares, C. Armero, A. Forte, and L. Blanch, "Bayesian joint modeling of bivariate longitudinal and competing risks data: an application to study patient-ventilator asynchronies in critical care patients," *Biometrical Journal*, vol. 59, no. 6, pp. 1184–1203, 2017.

[11] E. Waldmann, D. Taylor-Robinson, N. Klein et al., "Boosting joint models for longitudinal and time-to-event data," *Biometrical Journal*, vol. 59, no. 6, pp. 1104–1121, 2017.

[12] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," *Proceedings of the Thirteenth International Conference on Machine Learning Theory*, Morgan Kaufmann, San Francisco, 1996.

[13] C. Griesbach, A. Mayr, and E. Waldmann, "Extension of the Gradient Boosting Algorithm for Joint Modeling of Longitudinal and Time-to-Event Data Ar Xiv e-Prints," *Cornell University*, 2018.

[14] B. Hofner, "Variable selection and model choice in survival models with time-varying Effects diploma thesisLudwig-Maximilians-Universitat Munchen," *Open Access LMU*, 2008.

[15] G. Tutz and H. Binder, "Generalized additive modeling with implicit variable selection by likelihood- based boosting," *Biometrics*, vol. 62, no. 4, pp. 961–971, 2006.

[16] G. Tutz and F. Reithinger, "A boosting approach to flexible semiparametric mixed models," *Statistics in Medicine*, vol. 26, no. 14, pp. 2872–2900, 2007.

[17] G. Tutz and A. Groll, *Generalized Linear Mixed Models Based on Boosting in Statistical Modelling and Regression Structures*, K. Thomas and T. Gerhard, Eds., Springer-Verlag, Berlin Heidelberg, 2010.

[18] G. Tutz and A. Groll, "Regularization for generalized additive mixed models by likelihood-based boosting," *Methods of Information in Medicine*, vol. 51, no. 2, pp. 168–177, 2012.

[19] G. Tutz and A. Groll, "Likelihood-based boosting in binary and ordinal random effects models," *Journal of Computational and Graphical Statistics*, vol. 22, no. 2, pp. 356–378, 2013.

[20] A. Groll, *Variable selection by regularization methods for generalized mixed models*, Ludwig-Maximilians-Universitat Munchen Munich, Germany, 2011.

[21] A. Groll, *GMMBoost: Likelihood-Based Boosting for Generalized Mixed Models*, p. 1.1.3, 2020, *R package version*.

[22] B. Hofner, T. Hothorn, and T. Kneib, "Variable selection and model choice in structured survival models," *Computational Statistics*, vol. 28, no. 3, pp. 1079–1101, 2013.

[23] A. Mayr, H. Binder, O. Gefeller, and M. Schmid, "The evolution of boosting algorithms," *Methods of Information in Medicine*, vol. 53, no. 6, pp. 419–427, 2014.

[24] C. Griesbach, B. Säfken, and E. Waldmann, "Gradient boosting for linear mixed models," *The International Journal of Biostatistics*, 2021.

[25] C. Griesbach, A. Groll, and E. Bergherr, "Addressing cluster-constant covariates in mixed effects models via likelihood-based boosting techniques," *PLoS One*, vol. 16, no. 7, article e0254178, 2021.

[26] D. I. Abrams, A. I. Goldman, C. Launer et al., "A comparative trial of didanosine or zalcitabine after treatment with zidovudine in patients with human immunodeficiency virus

infection," *New England Journal of Medicine*, vol. 330, no. 10, pp. 657–662, 1994.

[27] H. Akaike, *Information Theory and the Extension of the Maximum Likelihood Principle in Second International Symposium on Information Theory: 267-281*, 1973.

[28] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, pp. 461–464, 1978.

[29] N. E. Breslow and D. G. Clayton, "Approximate inference in generalized linear mixed models," *Journal of the American Statistical Association*, vol. 88, pp. 9–52, 1993.

[30] L. Fahrmeir and G. Tutz, *Multivariate Statistical Modelling Based on Generalized Linear Models*, Springer-Verlag, New York, 2001.

[31] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)," *The annals of statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[32] P. Bühlmann and T. Hothorn, "Boosting algorithms: regularization, prediction and model fitting," *Statistical Sciences*, vol. 27, pp. 477–505, 2007.

[33] A. Groll, T. Hastie, and G. Tutz, "Selection of effects in Cox frailty models by regularization methods," *Biometrics*, vol. 73, no. 3, pp. 846–856, 2017.

[34] D. K. Lee, N. Chen, and H. Ishwaran, "Boosted nonparametric hazards with time-dependent covariates," *Annals of Statistics*, vol. 49, no. 4, 2021.