A General Machine Learning Framework for Survival Analysis

Andreas Bender(✉), David Rügamer, Fabian Scheipl, and Bernd Bischl

Department of Statistics, LMU Munich, Ludwigstr. 33, 80539 Munich, Germany
andreas.bender@stat.uni-muenchen.de

Abstract. The modeling of time-to-event data, also known as survival analysis, requires specialized methods that can deal with censoring and truncation, time-varying features and effects, and that extend to settings with multiple competing events. However, many machine learning methods for survival analysis only consider the standard setting with right-censored data and proportional hazards assumption. The methods that do provide extensions usually address at most a subset of these challenges and often require specialized software that can not be integrated into standard machine learning workflows directly. In this work, we present a very general machine learning framework for time-to-event analysis that uses a data augmentation strategy to reduce complex survival tasks to standard Poisson regression tasks. This reformulation is based on well developed statistical theory. With the proposed approach, any algorithm that can optimize a Poisson (log-)likelihood, such as gradient boosted trees, deep neural networks, model-based boosting and many more can be used in the context of time-to-event analysis. The proposed technique does not require any assumptions with respect to the distribution of event times or the functional shapes of feature and interaction effects. Based on the proposed framework we develop new methods that are competitive with specialized state of the art approaches in terms of accuracy, and versatility, but with comparatively small investments of programming effort or requirements for specialized methodological know-how.

Keywords: Survival analysis · Gradient boosting · Neural networks · Competing risks · Multi-state models

1 Introduction

Survival analysis is a branch of statistics that provides a framework for the analysis of time-to-event data, i.e., the outcome is defined by the time it takes until an event occurs. Analysis of such data requires specialized techniques because, in contrast to standard regression or classification tasks,

(a) the outcome can often not be observed fully (censoring, truncation),

F. Hutter et al. (Eds.): ECML PKDD 2020, LNAI 12459, pp. 158–173, 2021.
https://doi.org/10.1007/978-3-030-67664-3_10

(b) the features can change their value during the observation period (time-varying features (TVF),
(c) the association of the feature(s) with the outcome changes over time (time-varying effects (TVE)),
(d) one or more other events occur that make it impossible to observe the event of interest (competing risks (CR)),
(e) more generally, in a multi-state setting, observation units can move from and to different states (multi-state models (MSM)).

Failure to take these issues into consideration usually results in biased estimates, incorrect interpretation of feature effects on the outcome, loss of predictive accuracy or a combination thereof. In this work, we use a reformulation of the survival task to a standard regression task that provides a holistic approach to survival analysis. Within this framework, censoring, truncation and time-varying features (TVF) can be incorporated by specific data transformations and extensions to time-varying effects (TVE) as well as competing risks and multi-state models can be re-expressed in terms of interaction effects. This abstraction of the survival task away from specialized algorithms is illustrated schematically in Fig. 1. Task-appropriate pre-processing (leftmost subgraph) yields a standardized data format that allows the estimation of feature-conditional hazard rates using any learning algorithm that can minimize the negative Poisson log-likelihood, such as, GBT, deep neural networks (DNN), regularization based methods, and others (middle subgraph).

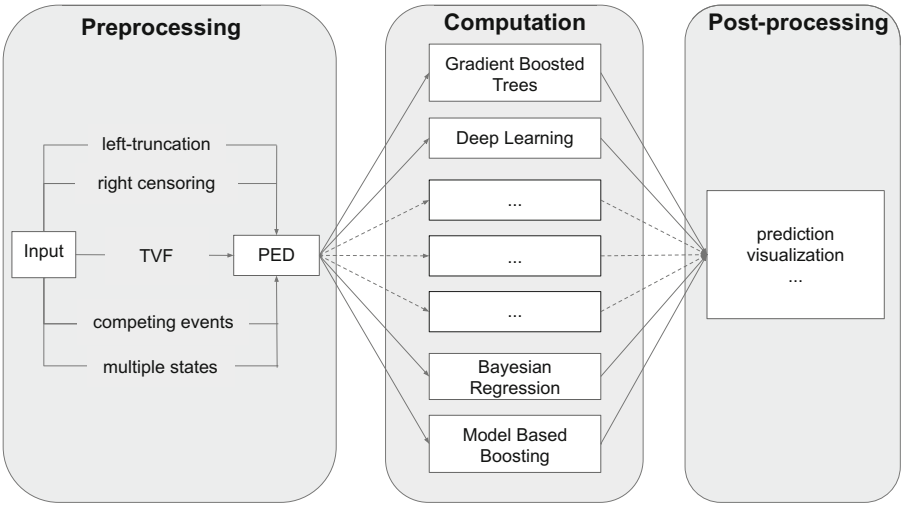

Fig. 1. An abstraction of survival analysis for different tasks. The structure of the piece-wise exponential data (PED) depends on the task requirements, e.g., left truncation or competing risks. Given the appropriate pre-processing, the estimation step is computationally independent of the survival task, except for an appropriate use of interaction terms.

Our Contributions
We define a general machine learning framework for survival analysis based on piece-wise exponential models (cf. Sect. 2). Within this framework, different concepts specific to time-to-event data analysis can be understood in terms of data augmentation and inclusion of interaction terms. By re-expressing the survival task as a Poisson regression task, a large variety of algorithms become available for survival analysis. Based on the proposed approach, we implement a gradient boosted trees algorithm with comparatively low development effort and show that it achieves state-of-the-art performance (cf. Sect. 3).

Related Work
The machine learning community has developed many highly efficient methods for high-dimensional settings in different domains, including survival analysis. The individual methods and implementations, however, often only support a subset of the cases relevant for time-to-event analysis mentioned above. For example, the random survival forest (RSF) proposed in [22] was later extended to the competing risks setting [21], but does not support left-truncation, TVF and TVE, or multistate models. Another popular implementation of random forests [35] only supports right censored data and proportional hazards models. An extension of RSF, the oblique RSF (ORSF, [23]) was shown to outperform other RSF based algorithms, but has the same limitations. With respect to TVF and TVE, a review of tree- and forest-based methods for survival analysis stated that "the modeling of time–varying features and time–varying effects deserves much more attention" [6]. Similarly, a more recent review of machine learning methods for survival analysis [34] only lists the time-dependent Cox model [25, Ch. 9.2], and L1- and L2-regularized extensions thereof, as a possibility for the inclusion of TVF.

Deep learning based methods for time-to-event data have also received much attention lately. An early use of neural networks for Cox type models was proposed by [10]. More recently, [31] presented a framework for deep single event survival analysis based on a joint latent process for features and survival times using deep exponential families. For competing risks data, a deep learning framework based on Gaussian processes was described in [1]. Another recent framework is DeepHit, which can handle competing risks using a custom loss function [28] and was extended to handle TVF [27], but did not discuss left-truncation, multistate models and TVE.

Boosting has also been a popular technique for high-dimensional survival analysis. For example, [5] propose a Cox-type boosting approach for the estimation of proportional sub-distribution hazards. A flexible multi-state model based on the stratified Cox partial likelihood in the context of model-based boosting [17] is presented in [32]. Furthermore, an implementation of gradient boosted trees (GBT) for the Cox PH model is also available for the popular XGBoost implementation [8], which was also shown to perform well compared with the ORSF [23]. Recently, [29] derived a custom algorithm for gradient boosted trees that support TVF and demonstrate that their inclusion improves predictive performance compared to boosting algorithms that don't take TVF into account.

Compared to methods based on Cox regression, few publications have developed methods based on the piece-wise exponential model, on which the framework proposed here is based. Among them is an early application of neural networks to survival analysis suggested in [30] and extended by [4]. The latter offers a general framework based on the representation of generalized linear models via feed forward neural networks, but does not discuss MSM. Piece-wise exponential trees with TVF and splits based on the piece-wise exponential survival function were suggested by [19]. A spline based estimation of the hazard function was discussed in [7], which could also be represented via neural networks (cf. [11]). A flexible estimation of piece-wise-exponential model based multi-state models with shared effects using structured fusion Lasso was developed in [33]. All of these methods can be viewed as special cases within the proposed framework. For example, [4] could be extended to different neural network architectures and MSMs, [19] could be extended to forests.

2 Survival Analysis as Poisson Regression

In the context of survival analysis, an observation usually consists of a tuple $(t_i, \delta_i, \mathbf{x}_i)$, where t_i is the observed event time for observation unit $i = 1, \ldots, n$, $\delta_i \in \{0, 1\}$ is the event- or status-indicator (i.e. 1 if event occurred, 0 if the observation of censored) and $\mathbf{x}_i$ is the p-dimensional feature vector. The presence of censoring requires special estimation techniques, as the time-to-event can not be observed when censoring occurs before the event of interest. Thus $t_i = \min(T_i, C_i)$, where T_i and C_i random variables of the event time and censoring time, respectively. A classic example is the time until death when censoring occurs as patients drop out of the study (unrelated to the event of interest, $T_i \perp C_i$). Left-truncation occurs when the event of interest already occurred before the subject could be included into the sample and thus presents a form of sampling bias. In some settings, another event could preclude observation of the event of interest or change the probability of its occurrence. In this case we speak of competing risks (CR), thus the observation consists of $(t_i, \delta_i, k, \mathbf{x}_i)$, where $k = 1, \ldots, K$ indicates the type of event that occurred at, t_i if $\delta_i = 1$. More generally, there might be multiple states that the observation units can transition from and to. We then speak of multi-state models (MSM) and k is an indicator for different transitions (cf. Eq. (8)).

In general, the goal of survival analysis is to estimate the conditional distribution of event times defined by the survival probability $S(t|\mathbf{x}) = P(T > t|\mathbf{x})$. While some methods focus on the estimation of $S(t|\mathbf{x})$ directly, it is often more convenient to estimate the (log-)hazard

$$\lambda(t|\mathbf{x}) := \lim_{\Delta t \to 0} \frac{P(t \leq T < t + \Delta t | T \geq t, \mathbf{x})}{\Delta t} \tag{1}$$

from which $S(t|\mathbf{x})$ follows as

$$S(t|\mathbf{x}) = \exp\left(- \int_0^t \lambda(s|\mathbf{x}) \mathrm{d}s \right). \tag{2}$$

Here we represent (1) via

$$\lambda(t|\mathbf{x}(t)) = \exp(g(\mathbf{x}(t), t)), \tag{3}$$

where g is a general function of potentially TVF $\mathbf{x}(t)$, that can include high-order feature interactions, non-linearity and time-dependence of feature effects (TVE) via an interaction with t.

In this work, we approximate (3) using the piece-wise exponential model [13]. Let t_i the observed event or censoring time and $\delta_i \in \{0, 1\}$ the respective censoring or event indicator for observation units $i = 1, \ldots, n$. The distribution of censoring times can depend on features but is assumed to be independent of the event time process T. By partitioning the follow-up, i.e., the time span under investigation, into $j = 1, \ldots, J$ intervals with cut-points $\kappa_0 = 0 < \cdots < \kappa_J$ and partitions $(\kappa_0, \kappa_1], \ldots, (\kappa_{j-1}, \kappa_j], \ldots (\kappa_{J-1}, \kappa_J]$, we can rewrite (3) using piece-wise constant hazard rates

$$\lambda(t|\mathbf{x}_i(t)) \equiv \exp(g(\mathbf{x}_{ij}, t_j)) := \lambda_{ij}, \quad \forall t \in (\kappa_{j-1}, \kappa_j], \tag{4}$$

with t_j a representation of time in interval j, e.g., $t_j := \kappa_j$ and $\mathbf{x}_{ij}$ the value of the TVF in interval j. Depending on the desired resolution, additional cut-points can be introduced at each time point at which feature values are updated, otherwise multiple feature values have to be aggregated in one interval. This model assumes that only the current value of $\mathbf{x}_{ij}$ affects the hazard in interval j, but more sophisticated approaches have been suggested within this framework that take into account the entire history of TVF [3]. Piece-wise constant hazards imply piece-wise exponential log-likelihood contributions

$$\ell_i = \log(\lambda(t_i; \mathbf{x}_i)^{\delta_i} S(t_i; \mathbf{x}_i)) = \sum_{j=1}^{J_i} \left(\delta_{ij} \log \lambda_{ij} - \lambda_{ij} t_{ij}\right), \tag{5}$$

where J_i is the last interval in which observation unit i was observed, such that $t_i \in (\kappa_{J_i - 1}, \kappa_{J_i}]$ and

$$\delta_{ij} = \begin{cases} 1 & t_i \in (\kappa_{j-1}, \kappa_j] \wedge \delta_i = 1 \\ 0 & \text{else} \end{cases}, \; t_{ij} = \begin{cases} t_i - \kappa_{j-1} & \delta_{ij} = 1 \\ \kappa_j - \kappa_{j-1} & \text{else} \end{cases}. \tag{6}$$

Concrete examples for the type of data transformations required to obtain (6) for right-censored data (including TVF) are provided in [2] (cf. Tables 1 and 2.

Using the working assumption $\delta_{ij} \overset{iid}{\sim} Poisson(\mu_{ij} = \lambda_{ij} t_{ij})$ and with $f(\delta_{ij})$ the Poisson density function, [13] showed that the Poisson log-likelihood

$$\ell_i = \log\left(\prod_{j=1}^{J_i} f(\delta_{ij})\right) = \sum_{j=1}^{J_i} (\delta_{ij} \log \lambda_{ij} + \delta_{ij} \log t_{ij} - \lambda_{ij} t_{ij}) \tag{7}$$

is proportional to (5) and therefore the former can be minimized using Poisson regression. Note that (7) can be directly extended to the setting with

left-truncated event times [16] by replacing $j = 1$ with j_i, the first interval in which observation unit i is in the risk set. The expectation is defined by $\mu_{ij} = \lambda_{ij} t_{ij} = \exp(g(\mathbf{x}_{ij}, t_j) + \log(t_{ij}))$. For estimation, $\log(t_{ij})$ is included as an offset, thus the hazard rate $\frac{\mu_{ij}}{t_{ij}} = \lambda_{ij} = g(\mathbf{x}_{ij}, t_j)$ is defined as the conditional expectation of having an event in interval j divided by the time under risk. Note that the Poisson assumption is simply a computational vehicle for the estimation of the hazard (4) rather than an assumption about the distribution of the event times. Despite the partition of the follow-up into intervals, this is a method for continuous event times as the information about the time under risk in each interval is contained in the offset and thus used during estimation. The number and placement of cut points controls the approximation of the hazard and could thus be viewed as a potential tuning parameter. In our experience, however, setting cut-points at the unique event times $\{t_i : \delta_i = 1, i = 1, \ldots, n\}$ in the training data always leads to a good approximation (at least with enough regularization) as the number of cut-points will increase in areas with many events. For larger data sets, however, we recommend to set these cut-points on a smaller representative sub-sample of the data set (cf. Sect. 4).

For the extension of (3) to MSMs, we define

$$\lambda(t|\mathbf{x}, k) = \exp\left(f(\mathbf{x}(t), t, k)\right), \; k = 1, \ldots, K, \tag{8}$$

as the transition specific hazard for the transition indexed by k, i.e., k is an index of transitions $m_k \to m_k'$ where m_k is the initial state and m_k' a transient or absorbing state. The set of possible transitions is given by $\{m_k \to m_k' : k = 1, \ldots, K\} \subseteq \{m \to m' : m, m' \in \{0, \ldots, M\}, m \neq m'\}$, where $M + 1$ denotes the total number of possible states. $f(\mathbf{x}(t), t, k)$ is a function of potentially time-varying features $\mathbf{x}(t)$, including multivariate and/or non-linear effects. The dependency of $f(\mathbf{x}(t), t, k)$ on time t (TVE) and transition k (MSM) is expressed in terms of interactions by defining $\tilde{\mathbf{x}} := (\mathbf{x}(t), t, k)$ and $f(\mathbf{x}(t), t, k) = f(\tilde{\mathbf{x}}(t))$. Let $t_{i,k}$ be the event or censoring time w.r.t. transition $m_k \to m_k'$ and $\delta_{i,k} \in \{0, 1\}$ the respective transition indicator. As extension of (6) we define

$$\delta_{ij,k} = \begin{cases} 1 & t_{i,k} \in (\kappa_{j-1}, \kappa_j] \wedge \delta_{i,k} = 1 \\ 0 & \text{else,} \end{cases}, t_{ij,k} = \begin{cases} t_{i,k} - \kappa_{j-1} & \delta_{ij,k} = 1 \\ \kappa_j - \kappa_{j-1} & \text{else} \end{cases}.$$

Table 1 shows how the data must be transformed in order to estimate (8) via PEMs for the competing risks setting, i.e., $k = 1, \ldots, K$ is an index of transitions $m_k = 0 \to m_k', m_k' \in \{1, \ldots, M\}$; a concrete example is given in Table 2. For each $i = 1, \ldots, n$, there is one row for each interval the observation unit was under risk for a specific transition. Thus, one data set is created for each transition such that transitions to state m_k' are encoded as 1 and everything else, i.e., censoring and transition to other states is encoded as 0. These transition-specific data sets, each containing a feature vector with the transition index k, are then concatenated. Note that we used the same interval split points κ_j for all transitions in Table 1. However, it would also be possible to choose transition specific cut-points $\kappa_{j,k}$, or, more generally, even use multiple time-scales

[20]. In the general multi-state setting, the number of observation units under risk might depend on the transition and the intervals visited by i are defined by $(t_{i,m}, \kappa_{j_{i,k}}], \ldots, (\kappa_{J_{i,k}-1}, \kappa_{J_{i,K}}]$, where $t_{i,m}$ is the time-point at which i enters state m.

Table 1. Data structure after transformation to the piece-wise exponential data format in the competing risks setting. Horizontal lines indicate a new observation unit $i = 1, \ldots, n$. Double horizontal lines indicate a new transition indexed $k = 1, \ldots, K$. As before, J_i is the interval in which i was observed last, i.e., $t_i \in (\kappa_{J_i-1}, \kappa_{J_i}]$. Features x depend on time via $x_{i,p}(t) = x_{ij,p}\ \forall t \in (\kappa_{j-1}, \kappa_j], p = 1, \ldots, P$. This is not a strict requirement, as additional split points could be set at each time point a feature value is updated. See Table 2 for a concrete example.

i	j	$\delta_{ij,k}$	t_j	t_{ij}	k	$x_{ij,1}$	$\ldots$	$x_{ij,P}$
1	1	$\delta_{11,1}$	t_1	t_{11}	1	$x_{11,1}$	$\ldots$	$x_{11,P}$
1	2	$\delta_{12,1}$	t_2	t_{12}	1	$x_{12,1}$	$\ldots$	$x_{12,P}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	J_1	$\delta_{1J_1,1}$	t_{J_1}	t_{1J_1}	1	$x_{1J_1,1}$	$\ldots$	$x_{1J_1,P}$
2	1	$\delta_{21,1}$	t_1	t_{21}	1	$x_{21,1}$	$\ldots$	$x_{21,P}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	1	$\delta_{n1,1}$	t_1	t_{n1}	1	$x_{n1,1}$	$\ldots$	$x_{n1,P}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	J_n	$\delta_{nJ_n,1}$	t_{J_n}	t_{nJ_n}	1	$x_{nJ_n,1}$	$\ldots$	$x_{nJ_n,P}$
1	1	$\delta_{11,2}$	t_1	t_{11}	2	$x_{11,1}$	$\ldots$	$x_{11,P}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	1	$\delta_{11,K}$	t_1	t_{11}	K	$x_{11,1}$	$\ldots$	$x_{11,P}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

In Sect. 3 we evaluate the suggested approach using an implementation based on GBT that we refer to as GBT (PEM). As a concrete computing engine we used the extreme gradient boosting (XGBoost) library [8] without any alterations to the algorithm. Therefore all features of the library can be used directly when estimating the hazard on the transformed data set. Note, however, that depending on the algorithm used, one must be able to specify an offset during estimation and potentially some other, algorithm or implementation specific settings. For example, when using XGBoost to estimate the GBT (PEM), the objective function needs to be set to the Poisson objective and the base score must be set to 1, because the default of 0.5 would imply a wrong offset, while $\log(1) = 0$. The offset ($\log(t_{ij})$) must be attached to the data via the base margin

Table 2. Data transformation for a hypotethical competing risks example ($K = 2$) for 3 subjects, $i = 1, \ldots, 3$. Subject $i = 1$ experienced an event of type $k = 2$ at $t_1 = 1.3$, subject $i = 3$ experienced an event of type $k = 1$ at time $t_3 = 2.7$, subject 2 was censored at $t_2 = 0.5$. Tables present the transformed data with intervals $(0, 1], (1, 1.5], (1.5, 3]$ for causes $k = 1$ (left) and $k = 2 = K$ (right). These can be used to estimate cause specific hazards, by applying the algorithm to each of the tables separately or cause specific hazards with potentially shared effects, by stacking the tables and using k as a feature.

i	j	δ_{ij}	t_j	t_{ij}	k
1	1	0	1	1	1
1	2	0	1.5	0.3	1
2	1	0	1	0.5	1
3	1	0	1	1	1
3	2	0	1.5	0.5	1
3	3	1	3	1.2	1

i	j	δ_{ij}	t_j	t_{ij}	k
1	1	0	1	1	2
1	2	1	1.5	0.3	2
2	1	0	1	0.5	2
3	1	0	1	1	2
3	2	0	1.5	0.5	2
3	3	0	3	1.2	2

argument during estimation. In contrast, for the prediction of the conditional hazard $\lambda(t|\mathbf{x}_i) = \lambda_{ij}$ based on new data points, the offset should be omitted, otherwise the algorithm will predict $\hat{\mu}_{ij} = \hat{\lambda}_{ij} \cdot t_{ij}$ (the expectation) instead of $\hat{\lambda}_j(\mathbf{x})$ (the hazard). When predicting the cumulative hazard or survival probability, however, the time under risk in each interval must be taken into account, such that $\hat{S}(t|\mathbf{x}_i) = \exp\left(-\int \hat{\lambda}(t|\mathbf{x}_i)\mathrm{d}t\right) = \exp\left(-\sum_{j=1}^{j(t)} \hat{\lambda}_{ij}\tilde{t}_j\right)$, where $j(t)$ indicates the interval for which $t \in (\kappa_{j-1}, \kappa_j]$ and $\tilde{t}_j = \min(\kappa_j - \kappa_{j-1}, t - \kappa_{j-1})$ is the time spent in interval j. A prototype implementation of the GBT (PEM) algorithm that takes these issues into account and also provides the necessary helper functions for data transformation, estimation, tuning, prediction and evaluation is provided at https://github.com/adibender/pem.xgb.

3 Experiments

We perform a set of benchmark experiments with real world and synthetic data sets, exclusively using openly and directly available data, including a subset of data sets from recent publications on oblique random survival forests (ORSF, [23]) and DeepHit [28]. DeepHit and ORSF both have been shown to outperform other approaches such as RSF [22], conditional forests [18], regularized Cox regression [12] and DeepSurv [31]. We compare our approach in benchmarks against the two algorithms, which are evaluated separately based on evaluation measures used in the respective publications to ensure comparability. All code to perform respective analyses as well as additional supplementary files are provided in a GitHub repository: https://github.com/adibender/machine-learning-for-survival-ecml2020.

The data sets used for single event comparisons are listed in Table 3. The "synthetic (TVE)" data set is created based on an additive predictor $g(\mathbf{x}, t) = f_0(x_0, t) \cdot 6 - 0.1 \cdot x_1 + f_2(x_2, t) + f_3(x_3, t)$, where f_0, f_2 and f_3 are bivariate, non-linear functions of the inputs (see code repository for details) and $x_0, \ldots, x_3$ feature columns comprised in $\mathbf{x}$. Additionally, 20 noise variables are drawn from the uniform distribution $U(0, 1)$.

For the comparison with ORSF, we use the Integrated Brier Score $\text{IBS}(\tau) = \frac{1}{\tau} \int_0^\tau \widehat{\text{BS}}(u, \hat{S}) \text{d}u$, where $\widehat{\text{BS}}(t, \hat{S})$ is the estimated Brier Score at time t weighted by the inverse probability of censoring weights [15] and $\hat{S}$ the estimated survival probability function of the respective algorithm. In addition, [23] report the time-dependent C-Index [14]. We only consider the IBS here as it measures calibration as well as discrimination, while the C-index only measures the latter. Note that the IBS depends on the specific evaluation time τ and different methods might perform better at different evaluation times. Therefore, we calculate the IBS for three different time-points, the 25%, 50% and 75% quantiles of the event times in the test data, in the following referred to as Q25, Q50 and Q75, respectively.

Table 3. Data sets used in benchmark experiments for comparison with ORSF.

	Name	N	P	Censoring
1	PBC	412	14	61.90
2	Breast	614	1690	78.20
3	GBSG 2	686	8	56.40
4	Tumor	776	7	51.70
5	Synthetic (TVE)	1000	24	∼33%

For comparison with DeepHit, we use the metabric data set (cf. [28]) for single event comparison as well as two CR data sets. The "MGUS 2" data is described in [26]. The "synthetic (TVE CR)" data set is simulated using an additive predictor identical to the one used for the "synthetic (TVE)" data simulation for the first cause. The predictor for the second cause, has a simpler structure $f_0(x_0, t) + 2 \cdot x_4 - .1 \cdot x_5$, however, with non-proportional baseline hazard with respect to $x_0 \in \{-1, 1\}$. The number of noise variables is limited to 10 for this setting. Here we report the weighted C-index alongside the weighted Brier Score as it was the main measure reported in [28]. The proposed GBT (PEM) approach for CR (cf. Sect. 2) is a cause specific hazards model, however, the parameters of both causes are estimated jointly and the hazards of both causes can have shared effects (see Fig. 2). The simulation setting, therefore, constitutes a difficult setup because there are no shared effects and optimization w.r.t. the first cause will favor parameters that allow flexible models while the optimization w.r.t. the second cause favors sparse models and thus parameters that would restrict flexibility.

Table 4. Data sets used for the comparison with DeepHit. MGUS2 and Synthetic (TVE CR) are data set with two competing risks and additional right-censoring.

Name	N	P	Censoring (%)
METABRIC	1981	79	55.20
MGUS 2	1384	6	29.6
Synthetic (TVE CR)	500	14	∼23%

3.1 Evaluation

We compare four algorithms, the non-parametric Kaplan Meier estimate (Reference) as a minimal baseline, the Cox proportional hazards model [9] (baseline for linear, time-constant effects), the Oblique Random Survival Forest (ORSF) [23] and DeepHit [28]. For each experimental replication for a specific data set, 70% of the data is randomly assigned as training data and the remaining 30% is used to calculate the evaluation measures at three time points Q25, Q50 and Q75. Algorithms are tuned on the training data using random search with a fixed budget and 4-fold cross-validation. Each algorithm is then retrained on the entire training data set using the best set of parameters before making final predictions on the test set. The random search consists of 20 iteration for each algorithm. For the GBT (PEM), we define the search space as follows (possible range in brackets): maximum tree depth $\{1, \ldots, 20\}$, minimum loss reduction [0, 5], minimum child weight $\{5, \ldots, 50\}$, subsample percentage (rows) [0.5, 1], subsample percentage of features in each tree [.5, 1], L2-regularization [1, 3]. The learning rate is set to 0.05 and number of rounds to 5000, with early stopping after 50 rounds without improvement. For the ORSF we tune the elastic net mixing parameter (0, 1), the parameter that penalizes complexity of the linear predictor in each node (0.25, 0.75), minimum number of events to split node $\{5, \ldots, 20\}$ and minimum observations to split node $\{10, 40\}$. For DeepHit, we use 50 random search iterations, where we search through {1,2,3,5} shared layers with {50, 100, 200, 300} dimensions, {1,2,3,5} cause-specific network layers with {50,100,200,300} dimensions, ReLU, eLU or Tanh as activation function in these layers, a batch size in {32,64,128}, a maximum of 50000 iterations, a dropout rate of 0.6 (taken from the original paper) and a learning rate of 0.0001. The network specific parameters α and γ are also chosen in accordance with the original paper and set to 1 and 0, respectively, while the network specific parameter β is varied in the random search with possible values in {0.1, 0.5, 1, 3, 5}.

3.2 Results

The results for the experiments based on single-event scenarios comparison with ORSF are summarized in Table 5. The proposed method performs well in many settings in comparison to ORSF. Notably, both algorithms are often not much better than the Cox PH models indicating that the PH assumption is not violated

strongly in those data sets and the sample size might be too small to detect small deviations w.r.t. to non-linearity of feature effects, interaction effects and time-varying effects. The "synthetic (TVE)" setting illustrates that in the presence of strong, non-linear and non-linearly TVE our approach clearly outperforms the other methods. For the PBC data we additionally ran an analysis including TVF with GBT (PEM). In this case, the inclusion of TVF resulted in a worse performance (IBS of 4.3 (Q25), 6.4 (Q50) and 9.2 (Q75)), which indicates that the inclusion of TVF lead to overfitting or that simple inclusion of the last observed value and carrying the last value forward is not appropriate in this setting.

Table 5. Results of benchmark experiments for single event data comparing GBT (PEM) with ORSF. Bold numbers indicate the best performance for each setting.

Data		Kaplan-Meier	Cox-PH	ORSF	GBT (PEM)
Breast	Q25	**1.9**	–	2.0	2.0
	Q50	4.1	–	**4.0**	**4.0**
	Q75	7.2	–	**6.7**	**6.7**
GBSG 2	Q25	3.1	3.1	**2.9**	3.0
	Q50	6.8	6.5	**6.2**	6.4
	Q75	12.5	11.4	**11.1**	11.3
PBC	Q25	5.4	**3.7**	4.0	3.8
	Q50	9.1	**5.3**	6.1	5.5
	Q75	14.0	8.1	8.6	**7.8**
Synthetic (TVE)	Q25	9.8	7.3	7.0	**4.6**
	Q50	19.2	10.3	9.9	**6.7**
	Q75	23.7	11.1	11.7	**8.6**
Tumor	Q25	6.7	6.0	**5.5**	5.8
	Q50	12.3	11.2	**10.8**	10.9
	Q75	17.6	16.3	**16.2**	**16.2**

Table 6 summarizes the results of comparisons with DeepHit. The GBT (PEM) again shows good overall performance. For the synthetic data set our method clearly outperforms the other approaches because it is capable of estimating non-linearity as well as time-variation. On the MGUS 2 data set, DeepHit shows the best performance for cause 1, while GBT (PEM) outperforms the other approaches for cause 2. On the synthetic data, the cause-specific Cox-PH model shows good discrimination (C-Index) for the second cause, but is worse than GBT (PEM) and DeepHit w.r.t. to the Brier Score.

Table 6. Results of benchmark experiments comparing GBT (PEM) with DeepHit for single event and CR data. Bold numbers indicate the best performance for each setting.

Data	Index	Method	Cause 1			Cause 2		
			Q25	Q50	Q75	Q25	Q50	Q75
METABRIC	Brier Score	Cox-PH	13.3	22.1	26.4	–	–	–
		DeepHit	14.3	23.5	27.0	–	–	–
		GBT (PEM)	**12.8**	**21.3**	**25.9**	–	–	–
	C-Index	Cox-PH	63.7	65.1	64.7	–	–	–
		DeepHit	68.6	63.3	54.9	–	–	–
		GBT (PEM)	**71.9**	**71.5**	**67.7**	–	–	–
MGUS 2	Brier Score	Cox-PH (CS)	23.6	43.7	64.3	13.4	20.5	22.3
		DeepHit	**22.8**	**41.0**	**57.8**	14.9	27.0	41.5
		GBT (PEM)	22.9	41.6	60.5	**13.0**	**20.1**	**22.1**
	C-Index	Cox-PH (CS)	66.7	**65.9**	**62.4**	68.8	69.4	70.1
		DeepHit	59.6	57.0	52.3	65.5	67.2	68.6
		GBT (PEM)	**68.4**	62.9	60.5	**72.6**	**70.9**	**70.8**
Synthetic (TVE, CR)	Brier Score	Cox-PH	9.4	13.1	25.1	35.5	44.3	50.6
		DeepHit	9.5	16.0	28.9	33.0	38.8	**41.0**
		GBT (PEM)	**7.2**	**11.6**	**20.6**	**30.1**	**38.0**	43.6
	C-Index	Cox-PH	90.2	89.5	85.4	**86.5**	**83.9**	**81.6**
		DeepHit	92.3	90.8	84.6	82.0	80.1	79.8
		GBT (PEM)	**93.9**	**92.2**	**87.5**	80.9	80.8	81.0

4 Algorithmic Details and Complexity Analysis

We now briefly describe algorithmic details and discuss the complexity of the resulting algorithms when using the proposed framework.

Algorithmic Details
The proposed framework is general in the sense that it transforms a survival task into a regression task. Nevertheless, different methods (and algorithms) have different strengths and weaknesses and different strategies can be applied to specify various alternative models within this framework. For example, in tree based methods, time-variation of feature effects could be controlled by allowing interactions of the time variable only with a subset of features, e.g. based on prior information, and similarly in order to control shared vs. transition specific effects in the multi-state setting. Tree-based methods are particularly intuitive when it comes to understanding the integration of TVE and extension to multi-state models via interaction terms into the model. This is illustrated in Fig. 2. For example, in panel (A) of Fig. 2, features and split points before the split w.r.t. time indicate feature effects common to all time-points. Once the data in panel (A) is split w.r.t. time t, the predicted hazard will be different for

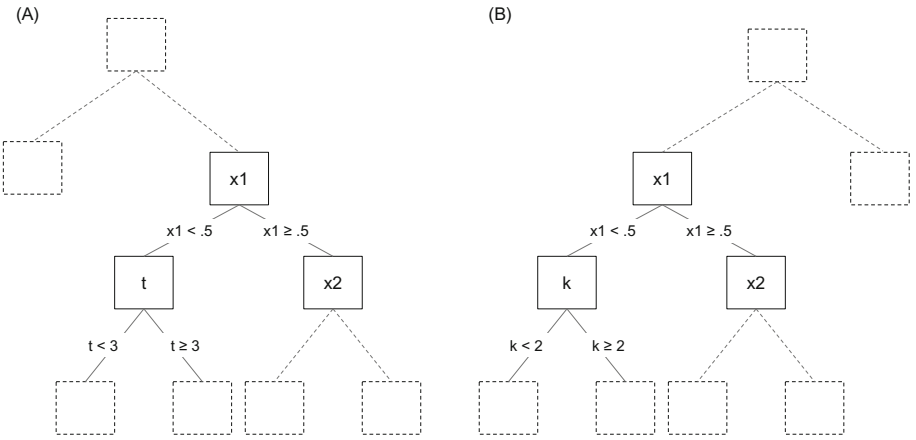

Fig. 2. Illustration of how TVE and shared vs. transitions specific effects can be understood in terms of feature interactions in tree based models.

intervals with $\kappa_j < 3$ and $\kappa_j \geq 3$ for observations with $x_1 < .5$. Similarly, in a multi-state setting (panel (B) in Fig. 2), splits above the split w.r.t. k indicate shared effects for all transitions, while splits below indicate different effects for transitions $k < 2$ vs. $k \geq 2$. Forcing a split w.r.t. to k at the root node would be equivalent to an estimation of cause specific hazards on each subset and no shared effects.

Neural networks are particularly flexible when it comes to the specification of different PEMs. For example, the network could be split in two subnetworks, one for the temporal component, one for features, which is equivalent to the specification of a proportional hazards model, while allowing for non-linearity and high-dimensional interactions in feature effects. Similarly, defining subnetworks of the time variable for each category of a categorical feature would imply a stratified proportional hazards model.

Complexity Analysis

As described in the literature review, various approaches exist that account for special survival characteristics like TVF, CR or continuous time-scale prediction by altering the underlying method. While adapting the structure of the algorithm itself potentially increases the complexity of the method, our approach leaves the algorithm of choice unchanged as different time points and transitions are simply included as features. This allows to employ commonly used prediction methods without introducing further algorithmic complexity. We note, however, that our approach might be improved upon in terms of scaling with respect to the number of intervals J relative to the number of observations n. In the worst case, the number of total data points is quadratic in n (or more precisely $\mathcal{O}(n(n+1)/2)$) when one interval cut-point is introduced for each observed event or censoring time. We therefore propose a refinement of the presented method that improves run-times without forfeiting performance. Instead of setting cut-points at all

unique event times, we suggest to define cut points more sparsely, for example, based on a sub-sample of the original data.

To investigate this strategy we conduct a scaling experiment where the sample size was consecutively doubled starting from $n = 400$ up to $n = 3200$. For each sample size, ten replications of one experiment as described for the "synthetic TVE" setting in Sect. 3 were performed and the elapsed time (hours) as well as performance (IBS) for two different strategies of cut-point selection was measured. The first strategy (full) uses all event times (t_i where $\delta_i = 1$) as cut-points. The second strategy (sub-sample) is equivalent to the first strategy, but event times were chosen based on a sub-sample of $n' = 200$, selected randomly from the training data in each iteration. Results in Table 7 show that the "sub-sample" strategy leads to an approximately linear increase in computation time while the performance remains virtually unchanged. Potentially, a sparser choice of cut-points could also lead to a more robust and thus improved hazard estimation, as more events are available in each interval, but we did not conduct a formal investigation in that regard.

Table 7. Results of the scaling experiments with n the number of observation in the simulated data. "strategy" refers to the way interval cut-points were selected with "full" splits at all unique event times (t_i where $\delta_i = 1$) and "sub-sample" refers to the selection of cut-points based on unique event times based on a random sub-sample of size $n' = 200$ (but all observations were used for estimation). Mean time (in hours) and IBS over 10 replications are reported for each setting.

	Strategy	N			
		400	800	1600	3200
Time (hours)	Full	0.10	0.48	2.49	8.94
	Sub-sample	**0.09**	**0.20**	**0.51**	**1.04**
IBS	Full	8.10	6.50	6.40	**5.90**
	Sub-sample	**8.00**	**6.40**	**6.20**	**5.90**

5 Conclusion

We have presented a general machine learning framework for time-to-event analysis based on a data augmentation strategy that reduces a large variety of survival analysis tasks to the optimization of a Poisson likelihood. We demonstrated its versatility and state-of-the-art performance. The availability of Poisson regression for most machine learning frameworks provides additional practical advantages. For example, photon-ML [37] is a scalable machine learning library for Apache Spark [36] that has no native support for survival analysis, but implements generalized linear mixed models. Therefore, survival modeling with high cardinality random effects (frailty) is directly available using our framework. Similarly, lightGBM [24], a high-performance implementation of GBT, currently has no implementation of survival methods, but could be also used for

high-dimensional survival tasks based on PEMs, including reliability analysis or churn analysis with intermediate states.

Acknowledgements. This work has been funded by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A. The authors of this work take full responsibilities for its content.

References

1. Alaa, A.M., van der Schaar, M.: Deep multi-task gaussian processes for survival analysis with competing risks. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 2326–2334 (2017)
2. Bender, A., Groll, A., Scheipl, F.: A generalized additive model approach to time-to-event analysis. Statistical Modelling p. 1471082X17748083 (2018)
3. Bender, A., Scheipl, F., Hartl, W., Day, A.G., Küchenhoff, H.: Penalized estimation of complex, non-linear exposure-lag-response associations. Biostatistics **20**(2), 315–331 (2018)
4. Biganzoli, E., Boracchi, P., Marubini, E.: A general framework for neural network models on censored survival data. Neural Netw. **15**(2), 209–218 (2002)
5. Binder, H., Allignol, A., Schumacher, M., Beyersmann, J.: Boosting for high-dimensional time-to-event data with competing risks. Bioinformatics **25**(7), 890–896 (2009)
6. Bou-Hamad, I., Larocque, D., Ben-Ameur, H.: A review of survival trees. Stat. Surv. **5**, 44–71 (2011)
7. Cai, T., Hyndman, R.J., Wand, M.P.: Mixed model-based hazard estimation. J. Comput. Graph. Stat. **11**(4), 784–798 (2002)
8. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 2016, pp. 785–794 (2016). arXiv: 1603.02754
9. Cox, D.R.: Regression models and life-tables. J. Royal Stat. Soc. Series B (Methodological) **34**(2), 187–220 (1972)
10. Faraggi, D., Simon, R.: A neural network model for survival data. Stat. Med. **14**(1), 73–82 (1995)
11. Fornili, M., Ambrogi, F., Boracchi, P., Biganzoli, E.: Piecewise exponential artificial neural networks (PEANN) for modeling hazard function with right censored data. In: Formenti, E., Tagliaferri, R., Wit, E. (eds.) CIBB 2013 2013. LNCS, vol. 8452, pp. 125–136. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-09042-9_9
12. Friedman, J.H., Hastie, T., Tibshirani, R.: Regularization paths for generalized linear models via coordinate descent. J. Stat. Softw. **33**(1), 1–22 (2010). number: 1
13. Friedman, M.: Piecewise exponential models for survival data with covariates. Ann. Stat. **10**(1), 101–113 (1982)
14. Gerds, T.A., Kattan, M.W., Schumacher, M., Yu, C.: Estimating a time-dependent concordance index for survival prediction models with covariate dependent censoring. Stat. Med. **32**(13), 2173–2184 (2013)
15. Gerds, T.A., Schumacher, M.: Consistent estimation of the expected brier score in general survival models with right-censored event times. Biometrical J. **48**(6), 1029–1040 (2006)

16. Guo, G.: Event-history analysis for left-truncated data. Sociol. Methodol. **23**, 217–243 (1993)
17. Hothorn, T., Bühlmann, P.: Model-based boosting in high dimensions. Bioinformatics **22**(22), 2828–2829 (2006)
18. Hothorn, T., Hornik, K., Zeileis, A.: Unbiased recursive partitioning: a conditional inference framework. J. Comput. Graph. Stat. **15**(3), 651–674 (2006)
19. Huang, X., Chen, S., Soong, S.j.: Piecewise exponential survival trees with time-dependent covariates. Biometrics **54**(4), 1420–1433 (1998)
20. Iacobelli, S., Carstensen, B.: Multiple time scales in multi-state models. Stat. Med. **32**(30), 5315–5327 (2013)
21. Ishwaran, H., et al.: Random survival forests for competing risks. Biostatistics **15**(4), 757–773 (2014)
22. Ishwaran, H., Kogalur, U.B., Blackstone, E.H., Lauer, M.S.: Random survival forests. Ann. Appl. Stat. **2**(3), 841–860 (2008)
23. Jaeger, B.C., et al.: Oblique random survival forests. Ann. Appl. Stat. **13**(3), 1847–1883 (2019)
24. Ke, G., et al.: LightGBM: a highly efficient gradient boosting decision tree. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 30, pp. 3146–3154. Curran Associates, Inc. (2017)
25. Klein, J.P., Moeschberger, M.L.: Survival Analysis: Techniques for Censored and Truncated Data. Springer, New York (2006)
26. Kyle, R.A., et al.: A long-term study of prognosis in monoclonal gammopathy of undetermined significance. N. Engl. J. Med. **346**(8), 564–569 (2002)
27. Lee, C., Yoon, J., Schaar, M.V.D.: Dynamic-DeepHit: a deep learning approach for dynamic survival analysis with competing risks based on longitudinal data. IEEE Trans. Bio-Med. Eng. **67**(1), 122–133 (2020)
28. Lee, C., Zame, W.R., Yoon, J., Schaar, M.V.d.: DeepHit: a deep learning approach to survival analysis with competing risks. In: Thirty-Second AAAI Conference on Artificial Intelligence (April 2018)
29. Lee, D.K.K., Chen, N., Ishwaran, H.: Boosted nonparametric hazards with time-dependent covariates. arXiv:1701.07926 [stat] (November 2019)
30. Liestbl, K., Andersen, P.K., Andersen, U.: Survival analysis and neural nets. Stat. Med. **13**(12), 1189–1200 (1994)
31. Ranganath, R., Perotte, A., Elhadad, N., Blei, D.: Deep Survival Analysis. arXiv:1608.02158 (August 2016)
32. Reulen, H., Kneib, T.: Boosting multi-state models. Lifetime Data Anal. **22**(2), 241–262 (2015). https://doi.org/10.1007/s10985-015-9329-9
33. Sennhenn-Reulen, H., Kneib, T.: Structured fusion lasso penalized multi-state models. Stat. Med. **35**(25), 4637–4659 (2016)
34. Wang, P., Li, Y., Reddy, C.K.: Machine learning for survival analysis: a survey. ACM Comput. Surv. (CSUR) **51**(6), 110:1–110:36 (2019)
35. Wright, M.N., Ziegler, A.: Ranger: a fast implementation of random forests for high dimensional data in C++ and r. J. Stat. Softw. **77**(1), 1–17 (2017)
36. Zaharia, M., et al.: Apache spark: a unified engine for big data processing. Commun. ACM **59**(11), 56–65 (2016)
37. Zhang, X., Zhou, Y., Ma, Y., Chen, B.C., Zhang, L., Agarwal, D.: Glmix: generalized linear mixed models for large-scale response prediction. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 363–372 (2016)