Semi-Structured Deep Piecewise Exponential Models

Philipp Kopper[1] PHILIPP.KOPPER@STAT.UNI-MUENCHEN.DE
Sebastian Pölsterl[2] SEBASTIAN.POELSTERL@MED.UNI-MUENCHEN.DE
Christian Wachinger[2] CHRISTIAN.WACHINGER@MED.UNI-MUENCHEN.DE
Bernd Bischl[1] BERND.BISCHL@STAT.UNI-MUENCHEN.DE
Andreas Bender[1*] ANDREAS.BENDER@STAT.UNI-MUENCHEN.DE
David Rügamer[1*] DAVID.RUEGAMER@STAT.UNI-MUENCHEN.DE
[1] *Chair of Statistical Learning and Data Science,*
Department of Statistics, LMU Munich,
Ludwigstr. 33, 80539 Munich, Germany
[2] *Artificial Intelligence in Medical Imaging,*
Department of Child and Adolescent Psychiatry, LMU Munich,
Waltherstr. 23, 80337 Munich, Germany

Abstract

We propose a versatile framework for survival analysis that combines advanced concepts from statistics with deep learning. The presented framework is based on piecewise exponential models and thereby supports various survival tasks, such as competing risks and multi-state modeling, and further allows for estimation of time-varying effects and time-varying features. To also include multiple data sources and higher-order interaction effects into the model, we embed the model class in a neural network and thereby enable the simultaneous estimation of both inherently interpretable structured regression inputs as well as deep neural network components which can potentially process additional unstructured data sources. A proof of concept is provided by using the framework to predict Alzheimer's disease progression based on tabular and 3D point cloud data and applying it to synthetic data.

Keywords: machine learning, deep learning, survival analysis, medical imaging, multimodal learning

1. Introduction

Similar to other fields, survival analysis has greatly benefited from the influx of machine learning (ML) and deep learning (DL) methods. However, especially early adaptations only focused on improved predictions of the event times or the survival probability in standard settings, i.e., right-censored, single-event data.

In this paper, we introduce a new method for continuous time-to-event data that combines advanced concepts from statistics and deep learning to form a new, versatile framework for survival analysis. The proposed framework enables the hazard-based learning of survival models via neural networks, which supports 1) survival tasks with right-censored, left-truncated, competing risks, or multi-state data; 2) estimation of inherently interpretable

[*] Contributed equally.

feature effects; 3) learning from multiple data sources (e.g., tabular and imaging data); 4) time-varying effects and time-varying features; 5) modeling of functional features (e.g., cumulative effects).

The new approach combines two general frameworks, the so-called piecewise exponential additive model (PAM) and semi-structured deep distributional regression (SDDR). The former provides a general framework for survival analysis that supports estimation of interpretable, additive effects of tabular features (referred to as the *structured* predictor in the following) and the latter embeds this structured part in a neural network (NN) while simultaneously explaining remaining heterogeneity or incorporating unstructured data (e.g., images or text) using additional NN components. Similar to the Cox model, our model does not require assumptions regarding the distribution of event times.

Before introducing the framework more formally in Section 2 and 3, we briefly recap the relevant literature and developments in relation to PAMs and neural network-based approaches to survival analysis.

Piecewise Exponential Additive Models The piecewise exponential additive model or PAM is an extension of the piecewise exponential model (PEM). The original formulation of the PEM, a parametric, linear effects, proportional hazards (PH) model, goes back to Holford (1980); Laird and Olivier (1981); Friedman (1982). The general idea is to partition the follow-up time into J intervals $(\kappa_{j-1}, \kappa_j]$, $j = 1, \dots, J$, and to assume piecewise constant hazards in each interval. Although the PEM was shown to be (mostly) equivalent to the Cox PH model (Whitehead, 1980), the latter prevailed in the survival literature, as the originally proposed PEM requires a careful choice of the number and placement of interval cut-points in a trade-off between the flexible estimation of the baseline hazards (large J) on the one hand and robustness of the estimates on the other hand (number of events per interval decreases as J increases).

Neural Network-based Approaches Neural network-based approaches for survival analysis also received a lot of attention. Early use of NNs for Cox type models was proposed by Faraggi and Simon (1995). More recently, various DL-based approaches have been proposed, most prominently: a single event survival model using deep exponential families (Ranganath et al., 2016), a framework based on Gaussian processes for competing risks data (Alaa and van der Schaar, 2017), and DeepHit a framework for discrete time-to-event data which can handle competing risks (Lee et al., 2018), which was recently extended to handle time-varying features (Lee et al., 2020).

The first combination of PEMs with a NN was proposed by Liestøl et al. (1994). Biganzoli et al. (2002) discussed the estimation of PEM by representing generalized linear models via feed-forward NNs, and Fornili et al. (2014) proposed the estimation of the shape of the hazard rate with NNs, rather than parametrically. Kvamme and Borgan (2019) also discussed the parametrization of the PEM via NNs with application to tabular data. Similar to the discussion above for structured PEMs, they found that the choice of cut-points is crucial for performance. As an extension of PAMs, the framework proposed in the following sections largely eliminates this problem.

The remainder of the paper is organized as follows: In Section 2 we will formally introduce the PAM, followed by its combination with SDDR in Section 3. As a proof of concept, Section 4 illustrates the application of the new framework to tabular and point cloud data with the goal to predict Alzheimer's disease progression and to synthetic data.

2. Piecewise Exponential Additive Models

2.1. General Model Definition

The general model specification of PAMs is given by

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

which defines the hazard h for time point $t \in \mathcal{T}$, conditional on a potentially time-varying feature vector $\mathbf{x}(t) \in \mathbb{R}^P$. The function $f(\cdot)$ represents the effect of (time-dependent) features $\mathbf{x}(t)$ on the hazard and can itself be potentially time- and transition-specific. k indicates a transition, e.g., from status 0 to status k in competing risks, or the transition between two states in the multi-state setting. We will omit the dependence on k for readability in the following and focus on survival tasks without competing risks or multiple states. Further omitting the dependence on t, (1) reduces to the familiar PH form known from the Cox model.

2.2. Data Transformation

PEMs and PAMs approximate (1) via piecewise constant hazards, which requires a specific data transformation, creating one row in the data set for each interval a subject was at risk. Assume n observations (subjects), $i = 1, \ldots, n$, for which the tuple $(t_i, \delta_i, \boldsymbol{x}_i)$ with event time t_i, event indicator $\delta_i \in \{0, 1\}$ (1=event, 0=censoring) and feature vector $\boldsymbol{x}_i$ being observed. PAMs partition the follow up into J intervals $(\kappa_{j-1}, \kappa_j]$, $j = 1, \ldots, J$. This implies a new status variable $\delta_{ij} = 1$ if $t_i \in (\kappa_{j-1}, \kappa_j] \wedge \delta_i = 1$, and 0 otherwise, indicating the status of subject i in interval j. Further, we create a variable t_{ij}, the time subject i was at risk in interval j, which will enter the analysis as an offset. Lastly, the variable t_j, (e.g., $t_j := \kappa_j$) is a representation of time in interval j and the feature based on which the model estimates the baseline hazard and time-varying effects. To transform data into the piecewise exponential data format (PED), time-constant covariates $\boldsymbol{x}_i$ are repeated for each of J_i rows, where J_i, denotes the number of intervals in which subject i was at risk. This data augmentation step transforms a survival task to a standard Poisson regression task. Depending on the censoring type, e.g., right-censoring, competing risks, or left truncation, the specifics of the data transformation vary, but the general principles remain the same. For more details refer to Bender et al. (2018, 2021).

2.3. Model Estimation

Given the transformed data, PAMs approximate (1) by

$$h(t|\mathbf{x}_i(t)) = \exp(f(\mathbf{x}_{ij}, t_j)) := h_{ij}, \forall t \in (\kappa_{j-1}, \kappa_j], \tag{2}$$

where $\mathbf{x}_{ij}$ is the feature vector of subject i in interval j.

Assuming $\delta_{ij} \sim \text{Poisson}(\mu_{ij} = h_{ij}t_{ij})$, the log-likelihood contribution of subject i is given by

$$\ell_i = \sum_{j=1}^{J_i} (\delta_{ij} \log(h_{ij}) - h_{ij}t_{ij}), \tag{3}$$

Equation (4) shows one standard parametrization of h_{ij} in the context of PAMs:

$$\log(h_{ij}) = \beta_0 + f_0(t_j) + \sum_{p=1}^{P} x_{ij,p}\beta_p + \sum_{l=1}^{L} f_l(x_{ij,l}), \tag{4}$$

with log-baseline hazard $\beta_0 + f_0(t_j)$, linear feature effects β_p and univariate, non-linear feature effects $f_l(x_{ij,l})$ of features $x_{ij,p}, x_{ij,l} \in \boldsymbol{x}_{ij}$. Both f_0 and f_l are defined via a basis representation, i.e.,

$$f_l(x_{ij,l}) = \sum_{m=1}^{M_l} \theta_{l,m} B_{l,m}(x_{ij,l})$$

with basis functions $B_{\cdot,m}(\cdot)$ (such as B-spline bases) and basis coefficients $\theta_{\cdot,m}$. To avoid underfitting, the basis dimensions M_0 (for f_0) and M_l (for f_l) are set relatively high. To avoid overfitting, the basis coefficients are estimated by optimizing an objective function, that penalizes differences between neighboring coefficients. Let $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_P)^\top$ and $\boldsymbol{\theta}_l = (\theta_{l,1}, \ldots, \theta_{l,M_l})^\top$, $l = 0, \ldots, L$. The objective function minimized to estimate PAMs is the penalized negative log-likelihood, which is given by

$$-\log \mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_L) + \sum_{l=0}^{L} \psi_l J(\boldsymbol{\theta}_l), \tag{5}$$

where the first term is the standard negative logarithmic Poisson likelihood, comprised of likelihood contributions (3), and the second term $J(\boldsymbol{\theta}_l)$ is a quadratic penalty with smoothing parameter $\psi_l \geq 0$ for the respective spline f_l. Larger ψ_l lead to smoother f_l estimates (see Wood, 2017; Bender et al., 2019, for details).

In the following, we will subsume all coefficients of structured additive model components such as $\boldsymbol{\beta}$ or $\boldsymbol{\theta}_l$ in the vector $\boldsymbol{w}$. All structured features can be furthermore summarized in a design matrix $\boldsymbol{B}$. This means we can represent the hazard as $\log(h_{ij}) = \boldsymbol{B}_{ij}\boldsymbol{w}$.

3. Semi-structured Learning with DeepPAM

DeepPAMs extend PAMs as defined in (1) by including unstructured model inputs into the additive predictor $f(\mathbf{x}(t), t)$. While PAMs restrict f to structured additive effects, the hypothesis space of DeepPAMs can also include G (deep) NN components $d_g(\cdot), g = 1, \ldots, G$. For readability, we assume only one NN component $d(\cdot)$ in the following. This NN predictor is used to process a potentially time-varying unstructured data source $\boldsymbol{z}(t)$. Different ways exist to integrate $d(\cdot)$ into PAMs, depending on the assumed relationship of structured and unstructured data. Here, we distinguish between two types of model classes, derived from the role of d. The first class of DeepPAMs assumes a time-constant effect of unstructured data sources

$$h(t|\boldsymbol{x}(t), \boldsymbol{z}(t)) = \exp\{f(\mathbf{x}(t), t) + d(\boldsymbol{z}(t))\}, \tag{6}$$

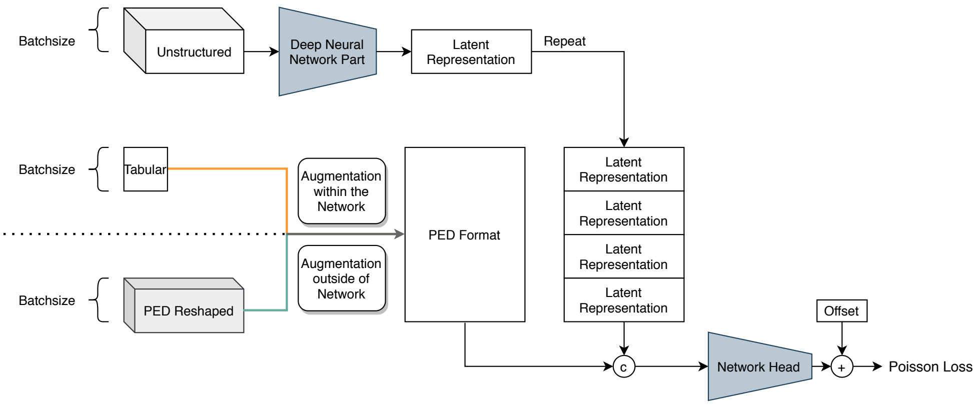

Figure 1: Architecture of a DeepPAM. The unstructured data, e.g., the images, are summarized to latent representations and combined with the tabular data in PED format using a concatenation (upper and right part of the image). The tabular data is either fed into the network in raw format and augmented within the network (orange path) or augmented outside of the network prior to the network training and reshaped to fit the defined batch size (green path). Latent representations of unstructured predictors are learned in a separate deep NN part and concatenated (c) with the transformed structured predictors by repeating the former. The joint influence of the latent representations and structured inputs is learned using one (or more) additional fully-connected layer(s) on top of the concatenated features (Network Head). Finally, the offset is added to the prediction and the network is trained using the Poisson loss.

which enriches the structured predictor of PAMs using one (or in general G) NN predictor(s). The second, more flexible version of DeepPAMs allows for an interaction between structured and unstructured predictors:

$$h(t|\boldsymbol{x}(t), \boldsymbol{z}(t)) = \exp\big\{f\left(\mathbf{x}(t), d(\boldsymbol{z}(t)), t\right)\big\},$$

where f now also depends on the unstructured data sources and the specified NN(s). As in standard PAMs, dependence on t allows deviations from the PH assumption.

Instead of combining PAMs with (deep) neural networks in a two-stage approach, we embed PAMs in a neural network using the idea of SDDR and train the network based on the (penalized) likelihood defined in (5). A general framework for end-to-end learning of *structured* (additive) predictors and neural network components has recently been proposed in Rügamer et al. (2020) and successfully applied in the context of transformation Baumann et al. (2020) and mixture Rügamer et al. (2020) models.

3.1. PED and Latent Representations

As outlined in section 2.2, the application of the framework requires a data transformation step. This brings many advantages, but one drawback is the increased data size. Thus, a major challenge when combining PED formatted data with unstructured data sources lies in the size and computational costs associated with each datum. The augmentation of tabular data can usually be done in memory. However, repeating unstructured data such as images for each interval of each observation requires multiple forward-passes of identical data through the network. This results not only in notably longer runtimes, but may also be redundant if the unstructured data sources are constant over time. For example, if we only have access to a patient's brain scan at the start of the study, it will not change for the remaining time intervals.

We address this issue in DeepPAMs using one of two options. Figure 1 schematically illustrates the two data handling pipelines within the network. Option 1 allows for PED format augmentation outside of the network. Instead of feeding the original tabular data into the network, the PED format is created prior to the network training and reshaped PED tensor batches with the same sampling dimension as the unstructured data source are passed through the network and then flattened internally to recover the original PED format. This option is especially useful when the network cannot deal with varying batch dimensions internally or the pre-processing has to be done outside the network for other reasons. The second option transforms the original tabular data into the PED format within the network. This can be more flexible as the PED format does not need to be transformed into a (ragged) tensor or must be padded as different subjects are usually observed for a varying number of intervals.

In both cases, we avoid repeating the original unstructured data source multiple times while still able to learn a common latent representation for the unstructured source for all time points of each subject. We achieve this by repeating the latent representation that is learned from the unstructured data source in the network. Finally, we combine these representations with the original tabular data.

In the following section, we will present a concrete architecture implementing this process together with the underlying Poisson loss optimization problem to learn a piecewise constant hazard from unstructured sources.

3.2. Learning piecewise Constant Hazards from Unstructured Data Sources

To make our concept of DeepPAM more concrete, we will now describe how the framework can be used to learn time-constant latent representations through an additive deep neural network predictor. This resembles the data generating process of the data we will later apply the framework to.

For simplicity assume a Q-dimensional unstructured data source $\boldsymbol{z}_i \in \mathbb{R}^{q_1 \times \ldots \times q_Q}$, e.g. an greyscale image with height q_1 and width q_2. $\boldsymbol{z}_i$ is passed through DeepPAM along with tabular inputs $\boldsymbol{x}_i$. If we assume that $\boldsymbol{z}_i$ has a piecewise constant effect on the hazard, i.e., an multiplicative effect on the baseline hazard, the corresponding DeepPAM is defined as in (6). While feeding the data into the network in batches defined on the subject level $i = 1, \ldots, n$, the resulting additive predictor on the level of the PED is given as follows:

$$h_{ij} = \exp\big\{\boldsymbol{B}_{ij}\boldsymbol{w} + \sum_{u=1}^{U} \zeta_{ij,u}\gamma_u\big\}.$$

Here, $\zeta_{ij,1}, \ldots, \zeta_{ij,U}$ are U latent representations learned from the unstructured data source $\boldsymbol{z}_i$ and $\gamma_1, \ldots, \gamma_U$ the corresponding feature effects. The γs are jointly learned with the weights $\boldsymbol{w}$ of the structured network part in a shared network head, which is a 1-unit hidden layer with linear activation. The corresponding architecture is visualized in Figure 1.

4. Numerical Experiments

In the following, we apply the proposed method using a real-world data set on Alzheimer's Disease (Section 4.1) and a synthetic data set (Section 4.2) in order to study the behavior of our model.

4.1. Learning the Risk of Alzheimer's Disease Progression

Alzheimer's disease (AD) is mostly diagnosed among elderly individuals from their 60s on. Early stages imply mild cognitive impairment (MCI). Patients suffering from MCI are not impaired in their daily life while they still face significant cognitive symptoms. Not all patients suffering from MCI eventually develop dementia symptoms (Petersen, 2011; Langa and Levine, 2014). In our experiments, we use the data from the Alzheimer's Disease Neuroimaging Initiative (ADNI; Jack Jr et al., 2008) to predict the progression of patients with MCI to dementia. Since 2003 ADNI has collected data of different modalities such as magnetic resonance imaging (MRI) and positron emission tomography (PET) and other biomarkers, clinical and neuropsychological assessments.

Using data from ADNI, our main goal is to study whether medical images have additional predictive capacity when also accounting for commonly used predictors such as non-image biomarkers.

Data We use a subset of the ADNI database from Pölsterl et al. (2019), including 397 individuals with MCI at their baseline assessment. MRI scans of the patient were processed using FreeSurfer (Fischl, 2012) and ShapeWorks (Cates et al., 2008) to eventually obtain smooth left hippocampi surfaces represented as point clouds. We downsample the point clouds to 1,024 coordinates, normalize the coordinates as proposed by Qi et al. (2017) and add noise on the point clouds used for generalizability (cf. Qi et al., 2017). Next to the medical images, we make also use of tabular clinical data including the level of education (years of education binned to 4 intervals), age, and sex. Furthermore, we account for relevant predictive biomarkers for the progression of AD: FDG-PET, AV45-PET, levels of beta-amyloid 42 peptides (Abeta), total tau protein (Tau).

As an additional feature derived from the images we use the left hippocampus volume. Literature finds a link between structural changes in the hippocampus and proceeding dementia (Wachinger et al., 2016; Gerardin et al., 2009; Frisoni et al., 2008). This motivates the use of hippocampi data in the analysis.

Model For the unstructured deep part of the model, we use a variation of the PointNet (Qi et al., 2017) with a reduced number of weights similar to the parametrization suggested in Pölsterl et al. (2019). Our PointNet comprises a series of shared multi-layer-perceptrons (MLP), defined as 1D convolutions with kernel size 1, with batch normalization and ReLu-activation, global max pooling , and a series of global MLPs. We use the PointNet to learn 400 latent global features of the left hippocampi shape, which we further process through MLPs to one latent representation. Details on the architecture can be found in Pölsterl et al. (2019).

For the structured model part, we use linear effects for biomarkers and the hippocampus volume, a one-hot encoded linear feature effect for education and a smooth B-spline representation of age with five equidistant knots. Finally, we use a smooth B-spline representation of the time in order to model the baseline hazard. For spline regularization, we use a quadratic penalty term with its smoothing parameter (ψ_0) taken from the baseline PAM (cf. Section 2.3 and Equation (5)). We use the PAM estimates for a warm start of the structured part of the network and optimize the model using Adam (Kingma and Ba, 2015) with learning rate of 0.001. We perform early stopping based on the validation data performance.

Evaluation We learn the underlying association of clinical features and point clouds with patients' risk to AD progression on 10 different train-validation-test splits. In each repetition, we use 319 observations for training, 39 for validation, and 39 for testing. We evaluate our model by comparing the results to a baseline PAM using the same information as in the structured part of our network and also compare our model to the wide-and-deep approach by Pölsterl et al. (2019). Additionally, we include a Cox PH model with the same specification as the baseline PAM. We use the integrated Brier Score (IBS; Graf et al., 1999) at the 25%, 50%, and 75% quartiles of the training data as our evaluation criterion to measure predictive performance.

Results We find that all approaches yield comparable predictive performance, although the DeepPAM appears to perform slightly better when evaluated at later time-points compared to W&D. In Table 1 none of the models outperforms the others substantially. In particular, neither of the neural network-based approaches are able to improve survival probability estimates by additionally taking medical images in the form of left hippocampi point clouds into account. Different architectures of the PointNet did not alter these findings notably.

	1st Quartile	Median	3rd Quartile
KM	**3.0** (2.0)	7.3 (3.8)	10.3 (4.5)
PAM	**3.0** (2.1)	**6.3** (3.4)	7.8 (3.4)
DeepPAM	3.1 (2.1)	6.4 (3.2)	7.8 (3.2)
Cox PH	**3.0** (2.1)	**6.3** (3.4)	**7.7** (3.4)
W&D	**3.0** (2.0)	6.8 (3.5)	8.6 (3.6)

Table 1: Averaged IBS (standard deviation in brackets) for the Kaplan-Meier (KM) estimator, PAM, DeepPAM, Cox-PH and Pölsterl et al. (2019) (W&D). Kaplan-Meier, Cox PH and PAM do not make use of the point clouds. The values have been multiplied by 100 for better readability.

In the following, we examine our model in a second, more controlled setup, which allows examining whether the previous null result is an artifact of the proposed model.

4.2. Synthetic Data

We generate synthetic data on the basis of the ModelNet10 data set (Wu et al., 2015), also containing point clouds. We enrich this data set by simulating additional tabular input features and finally simulate right-censored survival times.

Data and Simulation The ModelNet10 data set consists of point clouds of ten different items. We first reduce the initial data set to the first three items (classes 0, 1, and 2) to allow for a better visual inspection of the model. Each item is stored as a set of coordinates in three dimensions (the x-, y- and z-coordinate) and can have different lengths. To reduce computational costs without losing too much information, we downsample the data of each point cloud to 1,024 points. We add a small uniformly distributed error on each coordinate for training. We further normalize each point cloud as proposed by Qi et al. (2017). The final synthetic data set consists of 1,008 observations for training, 144 observations for validation, and 216 for testing.

In order to relate the point clouds to a simulated survival time, we treat the item label as one-hot encoded categorical features. We assign the dummies different linear effects on the log hazard. We further simulate two additional uniformly distributed features x_1, x_2. These features are defined to have a linear effect on the log hazard. Finally, we simulate a time-varying baseline hazard. The hazard, based on which the survival times are simulated is thus given as follows:

$$\begin{aligned} h_i(t) = \exp(&\beta_0 + f_0(t) + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \\ &\gamma_1 \Delta_{i,1} + \gamma_2 \Delta_{i,2}), \end{aligned} \tag{7}$$

where the log baseline hazard is $\beta_0 + f_0(t) = -0.5 - 0.1 \cdot (t-4)^2$ with coefficients $\beta_1 = -0.25$ and $\beta_2 = 0.3$. Δ_1 and Δ_2 are dummy-encoded representations of the indexed item. The linear item effects $\gamma_1 = 0.5$, $\gamma_2 = -1$. All survival times are administratively censored at $t = 10$. Additionally, we introduce right-censoring. The censoring time follows an exponential distribution with a rate of $\lambda = 0.02$. The challenging part of the synthetic data set is that the latent class information, $\Delta_i, i = 0, 1, 2$, is unknown to the model and can only be approximated through a linear effect of the learned point cloud representation. Thus, the defined model does not perfectly mimic the true data generating process but estimates a surrogate for the classes.

Model We model the data generating process using the proposed DeepPAM as described in Section 3.2. The structured model part is defined in accordance with the true data generating process. For the point clouds we use a similar architecture as described in the previous subsection, but with different numbers of units in each layer. We reduce the number of parameters in the unstructured part of the network to 51,585 in total and use L2 regularization instead of dropout for regularization. The model is trained for a maximum of 75 epochs using Adam with a learning rate of 0.001. We employ early stopping based on the model's performance on the validation set.

Evaluation For training, validation, and testing, we split the data into three parts with approximately 75%, 10%, and 15%, respectively. We repeat our experiments 10 times using

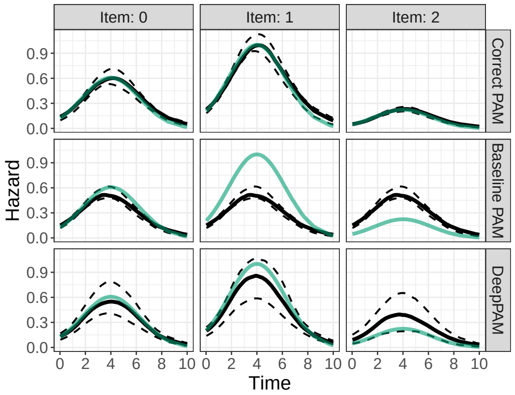

Figure 2: Median (black) and empirical 5%- and 95%-quantiles (dashed) of estimated hazard effects of DeepPAM. Effects are compared to the true effect (cyan) for each of three point cloud labels (columns) and each model (rows).

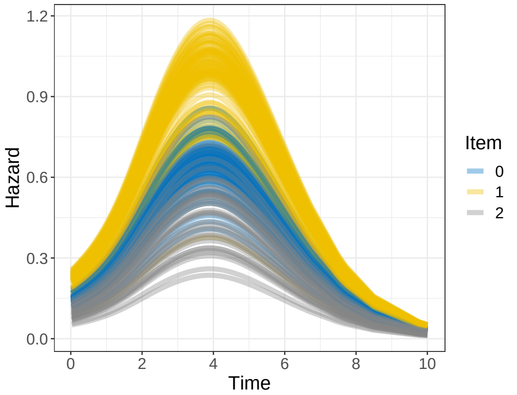

Figure 3: Example of the estimated baseline hazard effects by DeepPAM for different items (colors).

different simulated data. We evaluate effect estimates learned from point clouds using visual inspection and the model's predictive performance on a held-out test data set using the IBS.

We compare DeepPAM to a PAM model that is specified correctly w.r.t. to the data generating process (correct PAM) as well as a baseline PAM model, that does not include any point cloud information. The baseline model is used to check the model performance when no information from the point cloud items is taken into account and is expected to just average over the simulated linear item effects. The correct PAM is supplied with the (in reality) unknown correct label of the respective point cloud and should thus be able to recover the item-specific hazards.

Results Figure 2 shows the resulting estimates obtained from DeepPAM. Every single model plot corresponds to the distribution of 2,160 predicted hazard values for each latent item class, characterized by their median, their 5%- and 95%-quantile together with the true effect. As a comparison, we also plot the predictions of both PAMs. Figure 3 additionally displays the complete distribution of predicted latent effects for a single simulation run for all different items. Results indicate that the baseline model without additional class information predicts a constant hazard for all three classes. In contrast, DeepPAM is able to use the additional unstructured information and better discriminates between the three classes. The model recovers the effects of the two first classes very well. Although the effect of the third class is biased towards the mean effect, the true effect lies within the plotted quantile range. This means that our model behaves in the predicted behavior. All models tend to recover the linear coefficients β_1 and β_2 equally well.

IBS investigation support the previous results, with the true model having the highest and the baseline model the lowest IBS. Table 2 reports the performance of DeepPAM and the PAM baseline model relative to the performance of the ground truth model at the three quartiles of the survival time distribution. In addition, we also include the Kaplan-Meier estimator and the Cox PH model in the comparison. DeepPAM outperforms all models (except for the correctly specified PAM).

	1st Quartile	Median	3rd Quartile
KM	6.65 (3.25)	12.54 (5.47)	19.11 (6.87)
Cox PH	2.18 (1.60)	4.16 (3.03)	7.03 (4.07)
PAM	2.43 (1.79)	4.39(3.13)	7.22 (3.97)
DeepPAM	**0.97** (1.39)	**1.27** (1.83)	**2.41** (2.06)

Table 2: Averaged relative difference of the IBS (in %, standard deviation in brackets), compared to the correct PAM, for different quantiles (columns) and the different models (rows). We report the Kaplan-Meier estimator (KM), a Cox PH and a PAM model without point cloud information (Cox PH / PAM). The closer values are to the value 0, the more similar the performance of the model to the correct PAM.

5. Outlook

We present DeepPAM, a novel framework for survival analysis to seamlessly combine different data modalities. While the framework is very general, we demonstrate the use of DeepPAM to jointly model point clouds and tabular data. We show that DeepPAM yields comparable results to other benchmark models on an Alzheimer's disease prediction task. In this example, incorporating unstructured information in the model does not seem to improve predictions notably, but the DeepPAM yields slightly better results than the wide-and-deep approach by Pölsterl et al. (2019). We further demonstrate the efficacy of DeepPAM on a synthetic data set, where additional unstructured data information improves the model's predictions.

For future research, DeepPAM needs to be tested in more complex settings and applications to reinforce its value to time-to-event analysis. As the presented framework is flexible, various combinations of structured PAMs with (deep) neural networks are possible, in particular, deviations from the PH assumption and application to more complex survival tasks.

Acknowledgments

This work has been partly funded by the Bavarian State Ministry of Education, Science and the Arts (SP, CW) in the framework of the Centre Digitisation.Bavaria (ZD.B) and the German Federal Ministry of Education and Research (BMBF) under Grant Nos. 01IS18036A (PK, AB, BB, DR) and 031L0200A (SP, CW). The authors of this work take full responsibilities for its content.

References

Ahmed M Alaa and Mihaela van der Schaar. Deep multi-task gaussian processes for survival analysis with competing risks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 2326–2334. Curran Associates Inc., 2017.

Philipp Baumann, Torsten Hothorn, and David Rügamer. Deep conditional transformation models. *arXiv preprint arXiv:2010.07860*, 2020.

Andreas Bender, Andreas Groll, and Fabian Scheipl. A generalized additive model approach to time-to-event analysis. *Statistical Modelling*, 18(3):299–321, 2018. ISSN 1471-082X. doi: 10.1177/1471082X17748083. Publisher: SAGE Publications India.

Andreas Bender, Fabian Scheipl, Wolfgang Hartl, Andrew G. Day, and Helmut Küchenhoff. Penalized estimation of complex, non-linear exposure-lag-response associations. *Biostatistics*, 20(2):315–331, April 2019.

Andreas Bender, David Rügamer, Fabian Scheipl, and Bernd Bischl. A General Machine Learning Framework for Survival Analysis. In Frank Hutter, Kristian Kersting, Jefrey Lijffijt, and Isabel Valera, editors, *Machine Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science, pages 158–173, Cham, February 2021. Springer International Publishing. ISBN 978-3-030-67664-3. doi: 10.1007/978-3-030-67664-3_10.

Elia Biganzoli, Patrizia Boracchi, and Ettore Marubini. A general framework for neural network models on censored survival data. *Neural Networks*, 15(2):209–218, March 2002.

Joshua Cates, P Thomas Fletcher, Martin Styner, Heather Cody Hazlett, and Ross Whitaker. Particle-based shape analysis of multi-object complexes. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 477–485. Springer, 2008.

David Faraggi and Richard Simon. A neural network model for survival data. *Statistics in Medicine*, 14(1):73–82, 1995.

Bruce Fischl. Freesurfer. *Neuroimage*, 62(2):774–781, 2012.

Marco Fornili, Federico Ambrogi, Patrizia Boracchi, and Elia Biganzoli. Piecewise Exponential Artificial Neural Networks (PEANN) for Modeling Hazard Function with Right Censored Data. In Enrico Formenti, Roberto Tagliaferri, and Ernst Wit, editors, *Computational Intelligence Methods for Bioinformatics and Biostatistics*, Lecture Notes in Computer Science, pages 125–136, Cham, 2014. Springer International Publishing.

Michael Friedman. Piecewise exponential models for survival data with covariates. *The Annals of Statistics*, 10(1):101–113, 1982.

Giovanni B Frisoni, Rossana Ganzola, Elisa Canu, Udo Rüb, Francesca B Pizzini, Franco Alessandrini, Giada Zoccatelli, Alberto Beltramello, Carlo Caltagirone, and Paul M Thompson. Mapping local hippocampal changes in alzheimer's disease and normal ageing with mri at 3 tesla. *Brain*, 131(12):3266–3276, 2008.

Emilie Gerardin, Gaël Chételat, Marie Chupin, Rémi Cuingnet, Béatrice Desgranges, Ho-Sung Kim, Marc Niethammer, Bruno Dubois, Stéphane Lehéricy, Line Garnero, et al. Multidimensional classification of hippocampal shape features discriminates alzheimer's disease and mild cognitive impairment from normal aging. *Neuroimage*, 47(4):1476–1486, 2009.

Erika Graf, Claudia Schmoor, Willi Sauerbrei, and Martin Schumacher. Assessment and comparison of prognostic classification schemes for survival data. *Statistics in medicine*, 18(17-18):2529–2545, 1999.

Theodore R. Holford. The analysis of rates and of survivorship using log-linear models. *Biometrics*, 36(2):299–305, 1980.

Clifford R Jack Jr, Matt A Bernstein, Nick C Fox, Paul Thompson, Gene Alexander, Danielle Harvey, Bret Borowski, Paula J Britson, Jennifer L. Whitwell, Chadwick Ward, et al. The alzheimer's disease neuroimaging initiative (adni): Mri methods. *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 27(4):685–691, 2008.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Håvard Kvamme and Ornulf Borgan. Continuous and Discrete-Time Survival Prediction with Neural Networks. *arXiv preprint arXiv:1910.06724*, October 2019.

Nan Laird and Donald Olivier. Covariance analysis of censored survival data using log-linear analysis techniques. *Journal of the American Statistical Association*, 76(374):231–240, 1981.

Kenneth M Langa and Deborah A Levine. The diagnosis and management of mild cognitive impairment: a clinical review. *Jama*, 312(23):2551–2561, 2014.

Changhee Lee, William R. Zame, Jinsung Yoon, and Mihaela van der Schaar. DeepHit: A Deep Learning Approach to Survival Analysis With Competing Risks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, April 2018.

Changhee Lee, Jinsung Yoon, and Mihaela van der Schaar. Dynamic-DeepHit: A Deep Learning Approach for Dynamic Survival Analysis With Competing Risks Based on Longitudinal Data. *IEEE transactions on bio-medical engineering*, 67(1):122–133, 2020.

Knut Liestøl, Per Kragh Andersen, and Ulrich Andersen. Survival analysis and neural nets. *Statistics in Medicine*, 13(12):1189–1200, 1994.

Ronald C Petersen. Mild cognitive impairment. *New England Journal of Medicine*, 364 (23):2227–2234, 2011.

Sebastian Pölsterl, Ignacio Sarasua, Benjamín Gutiérrez-Becker, and Christian Wachinger. A wide and deep neural network for survival analysis from anatomical shape and tabular clinical data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 453–464. Springer, 2019.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

Rajesh Ranganath, Adler Perotte, Noémie Elhadad, and David Blei. Deep Survival Analysis. In *Proceedings of the 1st Machine Learning for Healthcare Conference*, volume 59, pages 101–114, 2016.

David Rügamer, Chris Kolb, and Nadja Klein. Semi-structured deep distributional regression: Combining structured additive models and deep learning. *arXiv preprint arXiv:2002.05777*, 2020.

David Rügamer, Florian Pfisterer, and Bernd Bischl. Neural mixture distributional regression. *arXiv preprint arXiv:2010.06889*, 2020.

Christian Wachinger, David H Salat, Michael Weiner, Martin Reuter, and Alzheimer's Disease Neuroimaging Initiative. Whole-brain analysis reveals increased neuroanatomical asymmetries in dementia for hippocampus and amygdala. *Brain*, 139(12):3253–3266, 2016.

John Whitehead. Fitting cox's regression model to survival data using glim. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 29(3):268–275, 1980.

Simon N. Wood. *Generalized Additive Models: An Introduction with R*. Chapman & Hall/Crc Texts in Statistical Science, Boca Raton, 2 rev ed. edition, June 2017.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.