SurviVAEl: Variational Autoencoders for Clustering Time Series

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Multi-state models are generalizations of time-to-event models, where individuals progress through discrete states in continuous time. As opposed to classical approaches to survival analysis which include only alive-dead transitions, states can be competing in nature and transient, enabling richer modelling of complex clinical event series. Classical multi-state models, such as the Cox–Markov model, struggle to capture idiosyncratic, non-linear, time dependent, or high-dimensional covariates for which more sophisticated machine learning models are needed. Recently proposed extensions can overcome these limitations, however, they do not allow for uncertainty quantification of the model prediction, and typically have limited interpretability at the individual or population level. Here, we introduce SurviVAEl, a multi-state survival framework based on a VAE architecture, enabling uncertainty quantification and interpretable patient trajectory clustering.

1 Introduction

1.1 Motivation

Survival analysis is of great importance in medicine with great interest in predicting the time to specific events like death or adverse events, while taking into account missing outcomes due to loss to follow-up. The de facto standard model in medicine today is the Cox proportional hazards (PH) model [8], a semi-parametric model making strong assumptions on the functional dependence of hazard rates on covariates. By now there has been a plethora of machine learning models generalizing beyond the PH assumption and allowing non-linear covariate influences [16, 20, 19, 5].

With the rise of electronic health records there is an ever increasing amount of information available and the possibility to model clinical patient trajectories in more detail, going beyond binary time-to-event analysis. In many cases, disease progression or other clinical events can be modeled with discrete states, e.g. tumor progression, side effects to treatments or relapse/remission after tumor surgery. The default approach in this case is the Cox–Markov model, describing any possible transition with a Cox-PH model without taking into account the previous disease trajectory and assuming piece-wise constant hazard rates. To advance beyond these very limiting assumptions survNode, a machine learning model based on neural ordinary differential equations [7], has been introduced by Groha et al. [12]. By using a neural network and introducing hidden states in the time evolution, survNode gets around the limiting assumptions of the Cox–Markov model, however, at the price of losing most of its interpretability as well as a clear means of uncertainty quantification. The goal of this manuscript is to add these two features, while retaining the flexibility of survNode.

1.2 Multi-state survival analysis

We define a continuous time stochastic process $\{Y(t); 0 \leqslant t \leqslant T\}$ over a finite state space $\mathsf{Y} = \{1, ..., S\}$, corresponding to the discrete states of the multi-state model. In the following, we will

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

briefly introduce the likelihood function and its relation to the Kolmogorov forward equations [18, 11]. As the approach introduced here is an extension of SURVNODE, the presentation closely follows that of [12].

For each patient, the process is observed at predefined times $t_1, \dots, t_m$ where the patient is in states $y(t_1), \dots, y(t_m)$. The likelihood is given by

$$P\left(y(t_1), \dots, y(t_m)|\mathcal{H}_{t_m};\theta\right) = P\left(y(t_1)\right) \prod\nolimits_{j=2}^{m} P_{y(t_{j-1})y(t_{j-1})}\left(t_{j-1}, t_j|\mathcal{H}_{t_{j-1}};\theta\right) \\ \times \lambda_{y(t_{j-1})y(t_j)}(t_j \mid \mathcal{H}_{t_j};\theta),$$

where θ denotes all free model parameters, $P\left(y(t_1)\right)$ the probability to be in the initial state, $P_{ij}\left(s,t\right)$ the transition probability between any set of states i, j occurring at time points s, t and $\mathcal{H}_t$ the filtration up until, but excluding time t. The full likelihood for all n patients is given by

$$\mathcal{L}(\theta;\{y_1, \dots, y_n\}) = \prod\nolimits_{i=1}^{n} P\left(y_i(t_1^i), \dots, y_i(t_{m_i}^i)|\mathcal{H}_{t_{m_i}};\theta\right),$$

with $y_i = \{y_i(t_1^i), \dots, y_i(t_{m_i}^i)\}, i = 1, \dots, n$. To accommodate censoring, the likelihood has to be adjusted. With the assumption of independent censoring, we observe $\{x_i, y_i, \delta_i; j = 1, \dots, m_i, i = 1, \dots, n\}$, where x_i are individual covariates, m_i is the number of transitions the individual i is going through and y_i are as above or the state at time of last contact. Censoring is indicated by $\delta_i = 0$ whereas we write $\delta_i = 1$ if the event is observed. The corresponding likelihood can then be written as

$$\mathcal{L}(\theta;\{y_1, \dots, y_n\}) = \prod\nolimits_{i=1}^{n} P\left(y_i(t_1^i)\right) \prod\nolimits_{j=2}^{m_{i-1}} P_{y_i(t_{j-1}^i)y_i(t_{j-1}^i)}\left(t_{j-1}^i, t_j^i;\theta\right) \lambda_{y_i(t_{j-1}^i)y_i(t_j^i)}(t_j^i \mid \theta) \\ \times P_{y_i(t_{m_i-1}^i)y_i(t_{m_i-1}^i)}\left(t_{m_i-1}^i, t_{m_i}^i;\theta\right) \left(\lambda_{y_i(t_{m_i-1}^i)y_i(t_{m_i}^i)}(t_{m_i}^i \mid \theta)\right)^{\delta_i},$$

where we omitted the dependence on the filtration for notational brevity. The evolution of the transition probabilities is governed by the Kolmogorov forward equation

$$\frac{\mathrm{d}P_{ij}(s,t|\mathcal{H}_t)}{\mathrm{d}t} = \sum\nolimits_k P_{ik}(s,t|\mathcal{H}_t)\lambda_{kj}(t|\mathcal{H}_t), \quad i = 1, \dots, S, \quad j = 1, \dots S, \tag{1}$$

which relates the intensities $\lambda_{ij}(t)$ to the transition probabilities $P_{ij}(s,t)$. Upon solving this equation we have all the ingredients for the likelihood function.

1.3 NeuralODE for multi-state survival modelling

The idea underlying SURVNODE is to model the instantaneous transition rates $\lambda_{ij}(t)$ using a neural network and solving the Kolmogorov forward equation using neural ODEs [7]. To incorporate the covariates and to include an approximation of the filtration of the process, SURVNODE introduces auxiliary memory states $m(t)$, itself modelled by an ODE

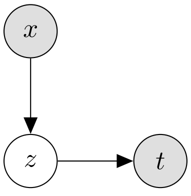
$$\frac{\mathrm{d}m_i}{dt} = M_i(t, \boldsymbol{P}(t), \boldsymbol{m}(t)).$$

The initial conditions are encoded by the covariates of the patient $m(0) = f(x)$, where f is given by a neural net. This implies having to solve the system of differential equations

$$\frac{\mathrm{d}P_{ij}(0,t)}{\mathrm{d}t} = \sum\nolimits_k P_{ik}(0,t)\lambda_{kj}(t, \boldsymbol{P(0,t)}(t), \boldsymbol{m}(t))$$
$$\frac{\mathrm{d}m_i}{\mathrm{d}t} = M_i(t, \boldsymbol{P}(0,t), \boldsymbol{m}(t)).$$

Using that $P(s,0) = P^{-1}(0,s)$ one can obtain $P_{ij}(s,t)$ at any s and t.

2 SurviVAEl



To obtain a quantification of model uncertainty and interpretability, we extend the model to a variational setting by introducing latent variables. Instead of maximum likelihood estimation, the objective will be the variational free energy or evidence lower bound ELBO. The variational model assumes the existence of a latent state z, which replaces the role of the initial memory state $\boldsymbol{m}(0)$ above, such that $\mathcal{L}(\theta;\mathcal{Y}, \boldsymbol{z})$ does not depend on the covariates $\boldsymbol{x}$ given $\boldsymbol{z}$. A graphical representation is given in Figure to the left.

Table 1: Benchmark on the METABRIC data-set. The other models are taken from [19] and [12].

Metric	Cox-PH[8]	DeepSurv [16]	DeepHit [20]	RSF [14]	survNode	SurviVAEl
c	0.628	0.636	0.675	0.649	0.667	0.646
ibs	0.183	0.176	0.184	0.175	0.157	0.170
inbll	0.538	0.532	0.539	0.515	0.477	0.503

We can derive the variational lower bound in a similar way to a (supervised) VAE [17, 15] as

$$\log p(t \mid x) = \log \int p(t \mid z)p(z \mid x) \geq E_{q(z|t,x)} \left[\log \frac{p(t \mid z)p(z \mid x)}{q(z \mid t, x)} \right].$$

The objective is then

$$\text{ELBO}(\theta, \mathcal{Y}) = \mathbb{E}_{q(\boldsymbol{z}|t,\boldsymbol{x})} \big[\log \mathcal{L}(\theta; \mathcal{Y}, \boldsymbol{z}) \big] - \mathcal{D}_{KL} \big(q(\boldsymbol{z} \mid t, \boldsymbol{x}) \parallel p(\boldsymbol{z} \mid \boldsymbol{x}) \big)$$

where we model the variational distribution $q(\boldsymbol{z}|t, \boldsymbol{x})$ and the prior $p(\boldsymbol{z}|\boldsymbol{x})$ as

$$q(\boldsymbol{z}|t, \boldsymbol{x}) = \mathcal{N}\big(\boldsymbol{z}; \mu_q(x,t), \text{diag}(\sigma_q^2(x,t))\big) \quad \text{and} \quad p(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}\big(\boldsymbol{z}; \mu_p(x), \text{diag}(\sigma_p^2(x))\big),$$

with neural networks for μ_q, μ_p, σ_q, and σ_p, encoding the covariates into the latent space. For prediction, we obtain realizations of the transition matrix $P_{ij}(0, t)$ by repeated sampling from the prior and taking the mean as well as the 95% credible interval.

3 Experiments

3.1 Survival: benchmark of model

We benchmark the variational model against various survival frameworks on the METABRIC breast cancer data set [9, 24]. We compare concordance (c)[2], integrated Brier score (ibs)[4], as well as the integrated binomial log-likelihood estimator (ibll)[19] using five-fold cross validation in Table 1. SurviVAEl is presented with only manual hyper-parameter tuning, whereas the other models went through an extensive hyperparameter search. We still find that the model is competitive in terms of survival prediction, retaining a lot of flexibility of survNode.

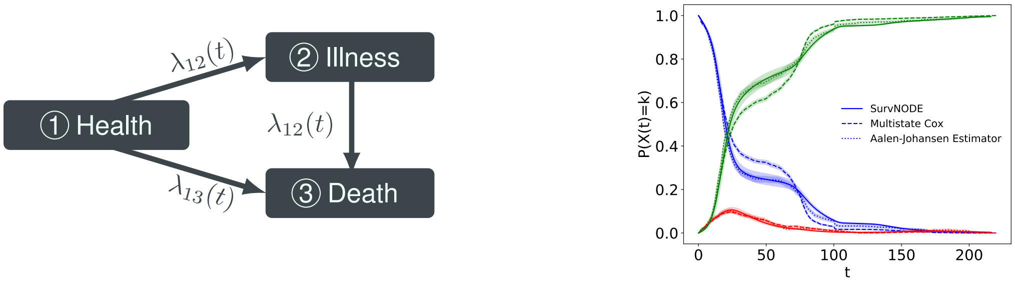


Figure 1: Left: Graphical representation of the Illness-death model. Right: Plot of probabilities for being in the different states of the illness-death model. Blue corresponds to “Health”, red to “Illness” and green to “Death”. The shaded area represents the 95% confidence intervals.

3.2 Multi-state survival

For illustration in the multi-state case, we compare performance on the population level with the non-parametric Aalen–Johansen estimator [1] for an illness-death model (see Figure 1). As a baseline, we simulate a data set with one binary covariate and proportional hazards violation using the **coxed** R package [13]. We compare our model with the standard tool in the multi-state survival literature, which is fitting a Cox proportional hazard model to each transition, treating the other events as censored [10]. The comparison can be seen in Figure 1, where we plot the predicted probabilities for the occupation of every state over time together with the non-parametric Aalen–Johansen estimator. We see a clear advantage of our model over the cause-specific Cox model and more importantly we see that the confidence intervals of SurviVAEl include the non-parametric estimator for most times.

Table 2: Comparison of SURVNODEand the Cox proportional hazard model in terms of calibration and concordance.

Model	calibration	concordance
Cox proportional hazards model [8]	0.250 ± 0.015	0.6244 ± 0.0064
SURVIVAEL	0.749 ± 0.071	0.6396 ± 0.0097

Calibration of the credible intervals While our model captures the non-parametric estimator by visual inspection, we seek to quantify the calibration performance in simulations where the ground truth is known. We simulate a survival data set with three covariates with the **`coxed`** package, where we also extract the underlying individual survival probabilities. To estimate calibration of the error intervals, we therefore calculate the average of fraction of times the true survival probabilities we sample from lie within the 95% credible interval. We compare the calibration of our model to the prediction from a Cox proportional hazards model using Wald type error estimates. For one random realization of the simulated data we perform a five fold cross validation in Table 2. We find that our model produces more consistent and better calibrated error intervals than the Cox proportional hazards model, as shown in Table 2

Clustering of the latent space An additional useful feature of the latent variable model can be found by inspection of the latent space of the model. Using a simulated illness-death model with nine covariates we run the variational SURVNODE model with early stopping using a validation set and then inspect the latent on a test data-set. Using UMAP [22] we identify five clusters (Figure 2). We examine the probabilities to be in each of the three states for each cluster in the validation data set using the non-parametric Aalen–Johansen estimator. As can be seen in Figure 2, the clusters are a meaningful unsupervised differentiation between patients and capture survival differences as well as differences in transitioning to the "Illness" state well. We can additionally obtain covariate effects associated with each cluster by using logistic regression. This feature has useful applications in a clinical setting, where identification of extreme survivors to a treatment while modeling other state transitions is of particular interest.

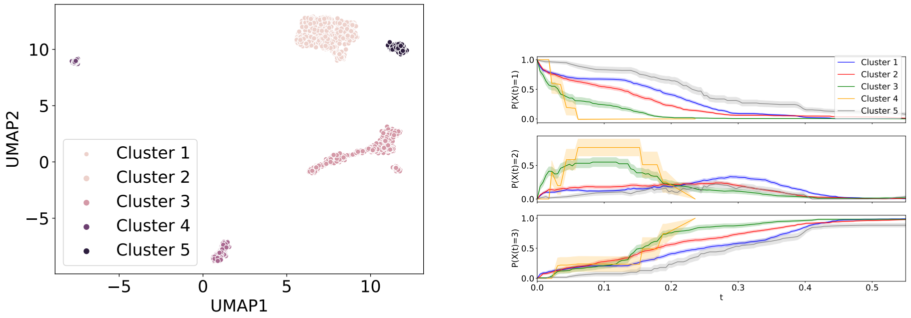


Figure 2: The latent space of the variational SURVNODE model shows meaningful clusters. The subset of patients in the clusters on the left are used in a non-parametric Aalen–Johansen estimator to obtain state occupation probabilities for all three states per cluster. The first panel is the probability for the "Health" state, the second for the "Illness" and the third for the "Death" state.

Our approach is directly applicable to survival analysis, where methods for example based on LDA [6] or variational inference [21] were recently proposed to cluster the latent space, but SURVIVAEL generalizes those to the multi-state setting.

4 Conclusion

We have introduced a variational framework for multi-state survival analysis based on neural ODEs and shown comparable performance in the special cases of survival. Our approach allows for the estimation of credible intervals and provides an interpretability aspect, which is absent from most machine learning survival methods and the first of its kind in the setting of multi-state models.

References

[1] Odd O Aalen and Søren Johansen. An empirical transition matrix for non-homogeneous markov chains based on censored observations. *Scandinavian Journal of Statistics*, pages 141–150, 1978.

[2] Laura Antolini, Patrizia Boracchi, and Elia Biganzoli. A time-dependent discrimination index for survival data. *Statistics in medicine*, 24(24):3927–3944, 2005.

[3] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017.

[4] Glenn W Brier and Roger A Allen. Verification of weather forecasts. In *Compendium of meteorology*, pages 841–848. Springer, 1951.

[5] Paidamoyo Chapfuwa, Chenyang Tao, Chunyuan Li, Courtney Page, Benjamin Goldstein, Lawrence Carin, and Ricardo Henao. Adversarial time-to-event modeling. *arXiv preprint arXiv:1804.03184*, 2018.

[6] Paidamoyo Chapfuwa, Chunyuan Li, Nikhil Mehta, Lawrence Carin, and Ricardo Henao. Survival cluster analysis. *Proceedings of the ACM Conference on Health, Inference, and Learning*, 4 2020. doi: 10.1145/3368555.3384465.

[7] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, pages 6571–6583, 2018.

[8] David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972.

[9] Christina Curtis, Sohrab P Shah, Suet-Feung Chin, Gulisa Turashvili, Oscar M Rueda, Mark J Dunning, Doug Speed, Andy G Lynch, Shamith Samarajiwa, Yinyin Yuan, et al. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486 (7403):346–352, 2012.

[10] Liesbeth de Wreede, Marta Fiocco, and Hein Putter. mstate: An r package for the analysis of competing risks and multi-state models. *Journal of Statistical Software, Articles*, 38(7):1–30, 2011. ISSN 1548-7660. doi: 10.18637/jss.v038.i07.

[11] W. Feller. On the theory of stochastic processes, with particular reference to applications. In *Proceedings of the [First] Berkeley Symposium on Mathematical Statistics and Probability*, pages 403–432, Berkeley, Calif., 1949. University of California Press.

[12] Stefan Groha, Sebastian M Schmon, and Alexander Gusev. Neural odes for multi-state survival analysis. *https://arxiv.org/abs/2006.04893*, 2020.

[13] Jeffrey J. Harden and Jonathan Kropko. Simulating duration data for the cox model. *Political Science Research and Methods*, 7(4):921–928, 2019. doi: 10.1017/psrm.2018.19.

[14] Hemant Ishwaran, Udaya B Kogalur, Eugene H Blackstone, Michael S Lauer, et al. Random survival forests. *The annals of applied statistics*, 2(3):841–860, 2008.

[15] Tom Joy, Sebastian Schmon, Philip Torr, Siddharth N, and Tom Rainforth. Capturing label characteristics in VAEs. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=wQRlSUZ5V7B`.

[16] Jared L Katzman, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network. *BMC medical research methodology*, 18(1):24, 2018.

[17] Diederik P Kingma, Danilo J Rezende, Shakir Mohamed, and Max Welling. Semi-supervised learning with deep generative models. *arXiv preprint arXiv:1406.5298*, 2014.

[18] Andrei Kolmogoroff. Über die analytischen methoden in der wahrscheinlichkeitsrechnung. *Mathematische Annalen*, 104(1):415–458, 1931.

[19] Håvard Kvamme, Ørnulf Borgan, and Ida Scheel. Time-to-event prediction with neural networks and cox regression. *Journal of Machine Learning Research*, 20(129):1–30, 2019.

[20] 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*, 2018.

[21] Laura Manduchi, Ričards Marcinkevičs, Michela C Massi, Thomas Weikert, Alexander Sauter, Verena Gotta, Timothy Müller, Flavio Vasella, Marian C Neidert, Marc Pfister, et al. A deep variational approach to clustering survival data. *arXiv preprint arXiv:2106.05763*, 2021.

[22] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018.

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[24] Bernard Pereira, Suet-Feung Chin, Oscar M Rueda, Hans-Kristian Moen Vollan, Elena Provenzano, Helen A Bardwell, Michelle Pugh, Linda Jones, Roslin Russell, Stephen-John Sammut, et al. The somatic mutation profiles of 2,433 breast cancers refine their genomic and transcriptomic landscapes. *Nature communications*, 7(1):1–16, 2016.

[25] Christopher Rackauckas and Qing Nie. Differentialequations.jl–a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software*, 5(1), 2017.

A Implementation details

All models are implemented in PyTorch [23] using the **torchdiffeq** package [7]. As our example networks are sufficiently small, we use backpropagation through the ODE solver to obtain gradients, however, using the adjoint method is of course possible as well. We use the `dopri5` method for the ODE solver with an absolute and relative tolerance of 10^{-8} in the ODE solver. To include the accuracy of the solution as a hyperparameter, we scale the event times to have the maximum value S, which we choose to be of $\mathcal{O}(1)$. To specify the non-zero elements of the transition rate matrix, a matrix with 1 indicators for non-zero off-diagonal elements and `NaN` indicators for all other elements are needed. We have the hyperparameters:

- Number of layers L_p and number of neurons per layer N_p with dropout p_p for multilayer perceptron for prior $p(z|x)$;
- Number of layers L_q and number of neurons per layer N_q with dropout p_p for multilayer perceptron for variational postierior $q(z|x, t)$;
- Number of layers L_Q and number of neurons per layer N_Q for multilayer perceptron modeling $\boldsymbol{Q}$;
- Number of latent states M;
- Coefficient of Lyapunov style loss term μ;
- ELBO parameter β
- Scaling coefficient for event times S;
- Learning rate l of the Adam optimizer;
- Weight decay w,

where the ELBO parameter β characterizes the relative weight between log-likelihood and Kullback-Leibler divergence, which we set to be 1 throughout the paper. Closer investigation of the clustering property with respect to this parameter would be of interest.

B Clustering: Covariates and survival strata

To further examine the clustering of the latent space, we can superimpose the nine binary covariates in the model on the UMAP projection. This can be seen in Figure 3. We see that some of the clusters clearly reflect the covariates, for example in the case of covariate one, which is the lowest third of the covariate with the largest effect size for one of the transitions in the simulation, we see that almost all the values are in one of the clusters. By characterizing the effect of the covariates on these clusters with specific survival properties, we can obtain the influence of the covariate on survival.

C Calibration of the credible intervals

A visual way to show the calibration of the credible intervals is to predict individual survival over time and plot together with the true underlying survival function obtained from the **coxed** R package. This can be seen in Figure 4. We see that the credible intervals contain the survival function in most of the cases.

D Experiments

D.1 Simulation of data

The simulated data in the publication is generated in two ways. First, we simulate data with the R package **coxed**.

In the survival cases, we choose three covariates, where one of the covariates has time varying coefficients to model a proportional hazards violation. We choose all coefficients to be of $\mathcal{O}(1)$, with a saw-tooth time dependence for the time dependent covariate. We sample 2048 patients for the training set and 1024 patients for the validation and test set respectively with event times between 0 and 100. In the case of the illness death model, we sample using the **coxed** package for every transition, assuming independence of each transition. We extract the covariates from the first sampled model and use them for the other two survival realizations, however choosing different coefficients.

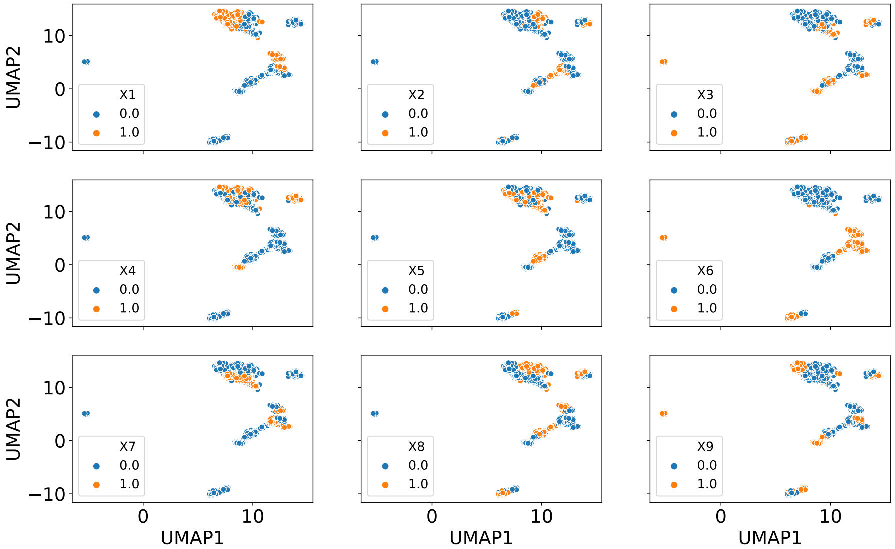


Figure 3: Possible values of the nine binary covariates in the model. We see that some of the clusters clearly reflect the covariates.

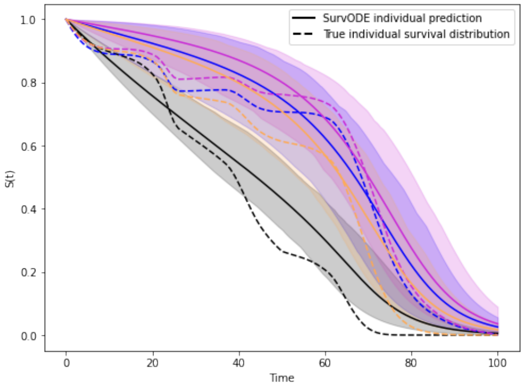


Figure 4: Predicted survival function vs real underlying survival function from the simulation. We see that the credible intervals cover the underlying survival function well.

Table 3: Characteristics of the METABRIC and SUPPORT data sets.

Data set	Size	Covariates	Unique Durations	Prop. Censored
SUPPORT	8873	14	1714	0.32
METABRIC	1904	9	1686	0.42

Due to a limitation of the **`coxed`** package, only the first sampled model can have time varying coefficients, with the other transitions then effectively being sampled from a Cox-model. In the competing case between "Illness" and "Death" from the "Health" state, we choose the first occurring time of the two sampled survival data realizations, no matter if there is censoring or not. The maximum time for the generated data in the competing case is $T = 100$, whereas we choose $T = 50$ for the transition from "Illness" to "Death".

The second way is to directly sample from a Markov-Jump process. For this we implement a Gillespie sampling algorithm in **`Julia`** [3], using the **`DifferentialEquations.jl`** [25] package. We sample parameters for a Weibull distribution for each transition in the multi-state case and multiplicatively add covariate dependence in a proportional hazards way. To break proportional hazards, we use time dependent coefficients for two of the 12 covariates, as in the above sampling algorithm. We choose all coefficients to be of $\mathcal{O}(1)$. We sample 5000 patients, which we then split into 64% training, 16% validation and 20% test set. As we specify the underlying hazard functions, ground truth for both hazard functions as well as probability distributions is directly accessible for any multi-state model. The simulation code is available on the SURVNODE github page.

D.2 Data sets and hyperparameters

The METABRIC and SUPPORT data sets are standard survival data sets for benchmarking. The characteristics are shown in 3 [19] and are obtained from the pycox python package [19]. The SYNTHETIC data set in the competing hazards case is taken from [20] and available on Github with 30000 patients and two outcomes, where 50% of patients experience any event, whereas the other 50% are censored.

For all benchmark experiments we do a five-fold cross validation where we split the data in an $80 - 20$ split into 20% test-data and the remaining data again in an $80 - 20$ split into 64% training data and 16% validation data. In the comparison with the non-parametric Aalen-Johansen estimator we used

- $L_p = 2$ with $N_p = 400$ and $p_p = 0.$;
- $L_q = 2$ with $N_p = 1000$ and $p_p = 0.$;
- $L_Q = 3$ with $N_Q = 1000$
- $M = 70$;
- $\mu = 10^{-4}$;
- $S = 1.$;
- $l = 1e - 4$;
- $w = 1e - 7$;
- $\beta = 1$,

and for clustering the latent space the hyperparameter setting we use is

- $L_p = 2$ with $N_p = 400$ and $p_p = 0.$;
- $L_q = 2$ with $N_p = 400$ and $p_p = 0.$;
- $L_Q = 2$ with $N_Q = 1000$
- $M = 50$;
- $\mu = 10^{-4}$;
- $S = 1.$;
- $l = 5e - 5$;
- $w = 1e - 7$;
- $\beta = 1.$

all of which were only manually hyperparameter tuned on train and validation set.