Deep Cox Mixtures for Survival Regression

Chirag Nagpal, Steve Yadlowsky, Negar Rostamzadeh and Katherine Heller

CMU, Google Brain

October 6, 2023

Presented by Mufan Wang

Introduction

The authors proposed a novel approach for Cox regression survival analysis using deep neural networks to learn mixtures of regressions.

- The paper addresses the limitations of the proportional hazards assumption in survival analysis models and introduces a mixture model that allows for flexible modeling of hazard ratios and latent group membership using deep neural networks.
- The proposed approach has the potential to improve the calibration of estimated conditional survival curves while maintaining excellent discriminative performance, making it valuable in medical decision-making and patient risk assessment.
- The experiments conducted by the authors involve multiple real-world datasets and examine the mortality rates of patients across different ethnicities and genders.

Introduction

Potential Applications

- The proposed approach of Deep Cox Mixtures for survival analysis regression models can be applied to clinical data to estimate patients' likelihood of survival characteristics based on biological measurements and demographic information.
- By improving calibration across different demographics, including underrepresented minority populations, the approach aims to reduce the miss-estimation of risk and improve the equity of models in healthcare decision-making.
- The approach is highly relevant to my project which is building model to predict the mortality and hospitalization rate of the End Stage Renal Disease (ESRD) patients.

Background

The **Expectation-Maximization (EM)** algorithm is a statistical method for finding maximum likelihood estimates of parameters in probabilistic models, where the model depends on unobserved latent variables.

- Divide the optimization problem into two simpler tasks:
  1. **Expectation (E) Step:** Calculate expected values based on the current estimates of the parameters.
  2. **Maximization (M) Step:** Maximize the expected likelihood to update parameter estimates.
- Iterate between these two steps until convergence.

Background

Given a likelihood function $L(\theta; X)$, where:

- θ = parameters of the model
- X = observed data

EM iteratively maximizes the following Q-function:

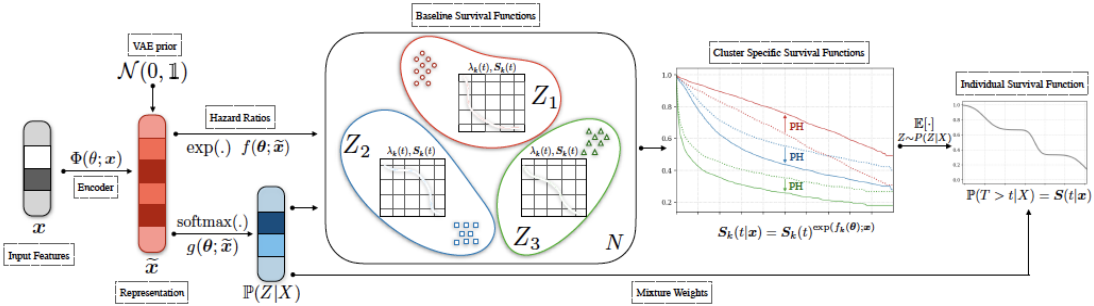
$$Q(\theta, \theta^{(t)}) = \mathbb{E}_{Z|X,\theta^{(t)}} [\log L(\theta; X, Z)]$$

where:

- Z = latent variables
- $\theta^{(t)}$ = current estimate of the parameters

The EM algorithm provides a structured approach to iteratively estimate model parameters in the presence of latent variables. It simplifies complex problems into manageable subproblems by alternating between expectation and maximization steps.

Methods


VAE prior
$\mathcal{N}(0, \mathbb{1})$
$\Phi(\theta; x)$
Encoder
x
Input Features
$\widetilde{x}$
Representation
Hazard Ratios
$\exp(.)\ f(\theta; \widetilde{x})$
softmax(.)
$g(\theta; \widetilde{x})$
$\mathbb{P}(Z|X)$
Baseline Survival Functions
$\lambda_k(t), S_k(t)$
Z_1
Z_2
Z_3
N
Cluster Specific Survival Functions
PH
PH
PH
$S_k(t|x) = S_k(t)^{\exp(f_k(\theta); x)}$
$\mathbb{E}[\cdot]$
$Z \sim P(Z|X)$
Individual Survival Function
$\mathbb{P}(T > t|X) = S(t|x)$
Mixture Weights

Methods

We consider a dataset of right censored observations $g = \{(z_i, \delta_i, y_i)\}_i$ of three tuples, where z_i are the covariates of an individual i, δ_i is an indicator of whether an event occurred or not and t_i is either the time of event or censoring as indicated by δ_i.
MLE based approach to learning $g(t|x) = P(T > t|X = x)$ from the data. Recall that the survival distribution $S(t|x)$ is isomorphic to the cumulative hazard function $\Lambda(t|x)$, and under continuity, this is equivalent to the hazard function $h(t|x)$. As a result, we will refer them in the parameters of the likelihood L that calculate explicitly those two models depending on information rate and gender when estimating outcomes. If there are strong reasons to believe that such information does not cause the outcome, other definitions of algorithmic fairness might be valid.

Methods

Interchangeably, Lin (2007) shows that the likelihood of the observed data g, up to constant factors, is

$$\mathcal{L}(\Lambda) = \prod_i (\lambda(z_i, y_i)^{\delta_i} S(t_i, z_i)).$$

In the following sections, we show how plugging in specific functional forms for $S(t|x)$ allows us to derive useful function estimation.

Methods

The key idea behind the Cox model is to assume that the conditional hazard of an individual, is $h(t|x) = \lambda_0(t)\exp(f(\theta, x))$, where f is typically a linear function. Under the Cox model, the full likelihood is in equation 1.

$$\mathcal{L}(\theta, \Lambda_0) = \prod_i \left(\lambda_0(t_i)\exp(f(\theta, x_i))\right)^{\delta_i} S(t_i)^{\exp(f(\theta, x_i))} \tag{1}$$

(2)

Likelihood estimate θ by maximizing the partial likelihood, $\mathcal{P}(\theta)$ defined below, and using the following estimate of b of the baseline survival function $S_0(t)$.

Methods

In the case of DCM we propose an extension to the Cox model, modeling an individual's survival function using a finite mixture of K Cox models, with the assignment of an individual i to each latent group mediated by a gating function $g()$. The full likelihood for this model is

$$\mathcal{L}(\theta, \Lambda_k) = \prod_{i=1}^{|\mathcal{D}|} \int_Z \left(\lambda(u_i; x_i)\right)^{\delta_i} S_k(u_i; x_i) P(Z = k|x_i)\, dZ$$

where,

$$\begin{aligned} \lambda(u_i; x_i) &= \Lambda_k(u_i) \exp\left(f_k(\theta, x_i)\right), \\ S_k(u_i; x_i) &= S_k(u_i) \exp\left(f_k(\theta; x_i)\right), \\ P(Z = k|X = x_i) &= \text{softmax}\left(g(\theta; x_i)\right). \end{aligned}$$

Architecture

We allow the model to learn representations for the covariates z by passing them through a encoding neural network, $f() : \mathbb{R}^d \to \mathbb{R}^s$. This representation then interacts with linear functions f and g defined on R^s, that determine the log hazard ratio and the mixture weights respectively.
The set of parameters for the encoder and the linear functions f and g are jointly denoted as θ. We experiment with a simple feed forward MLP and a variational auto-encoder for $f()$.
The parameters of the MLP and the VAE are learnt jointly during learning. For the VAE variant the encoder and the decoder architecture is kept the same. We also experiment with a variant that doesn't use representation learning and thus the functions f and g are linear and restricted to operate on the original features x. Figure 1 provides a schematic description of our approach.

Methods

Algorithm 1: Learning for DCM

Input : Training set, $\mathscr{D} = \{(\boldsymbol{x}_i, t_i, \delta_i)_{i=1}^N\}$;
batches, B;

while *<not converged>* **do**
  for $b \in \{1, 2, ..., B\}$ **do**
    $\mathscr{D}_b \leftarrow$ `sampleMiniBatch` $(\mathscr{D})$
    $\{\gamma_i\}_{i=1}^B \leftarrow$ **E-Step(** $\boldsymbol{\theta}, \{\widetilde{\boldsymbol{S}}_k\}_{i=1}^K$ **)**
    $\{\zeta_i\}_{i=1}^B \sim$ Categorical(γ)
    $\boldsymbol{\theta} \leftarrow$ **M-Step**$(\boldsymbol{\theta}, \{\zeta_i, \gamma_i\}_{i=1}^B)$
    for $k \in \{1, 2, ..., K\}$ **do**
      $\widehat{\boldsymbol{S}}_k \leftarrow$ `breslow`$(\boldsymbol{\theta}, \{(t_i, \delta_i)\}_{i=1;\zeta_i=k}^{|\mathscr{D}|})$
      $\widetilde{\boldsymbol{S}}_k \leftarrow$`splineInterpolate`$(\widehat{\boldsymbol{S}}_k)$
    end
  end
end

Return : learnt parameters, $\boldsymbol{\theta}$;
baseline survival splines $\{\widetilde{\boldsymbol{S}}_k\}_{i=1}^K$

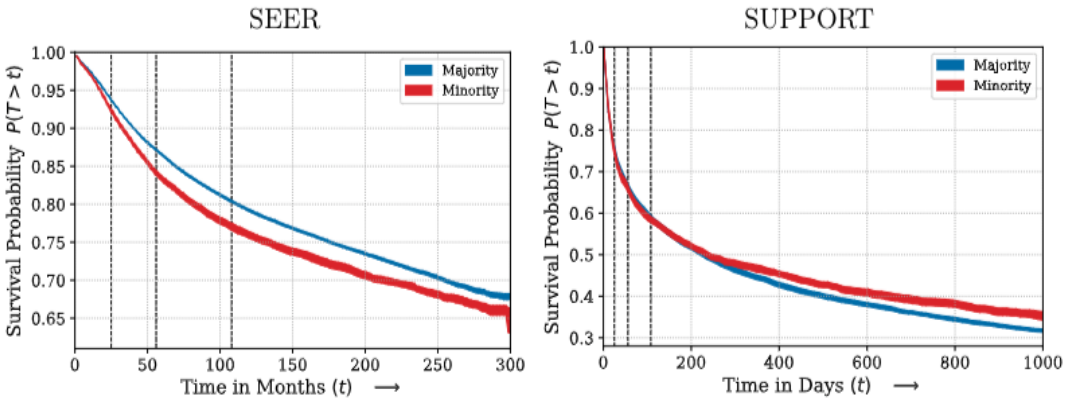
Experiments

The paper experiments on 3 real world, publicly available survival analysis datasets:

Table 1: Summary statistics for the datasets used in the experiments.

Dataset	N	d	Censoring (%)	Minority Class (%)	Event Quantiles		
					$t = 25$th	$t = 50$th	$t = 75$th
SUPPORT	9,105	44	31.89%	Non-White (21.02%)	14	58	252
FLCHAIN	6,524	8	69.93%	Female (44.94%)	903.25	2085	3246
SEER	55,993	168	72.82%	Non-White (23.77%)	25	55	108

Experiments



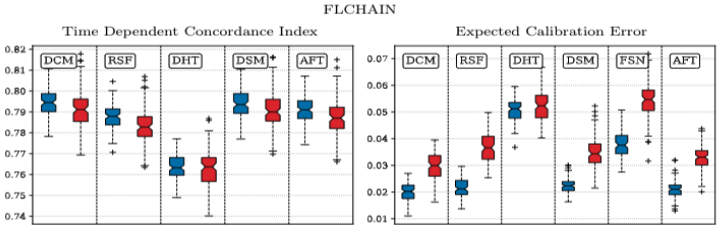
Base survival rates for the majority (White) vs. the other demographics in the SEER dataset estimated with a Kaplan-Meier estimator. Notice that the baseline survival rates differ across groups. Dashed lines respresent the 25th, 50th and 75th event quantiles.

Experiments

Baselines

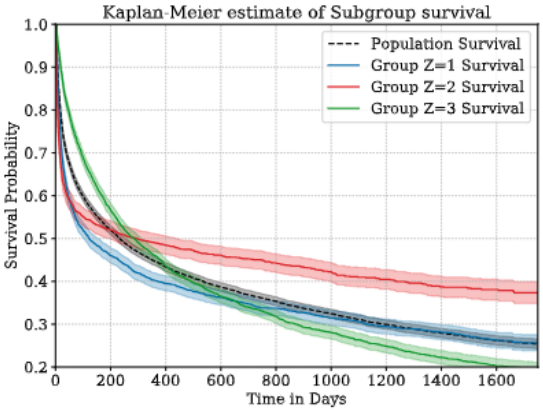
- Accelerated Failure Time (AFT)
- Deep Survival Machines (DSM)
- Deep Hit (DHT)
- Cox Proportional Hazards (CPH):
- Faraggi-Simon Net (FSN)/DeepSurv
- Random Survival Forest (RSF)

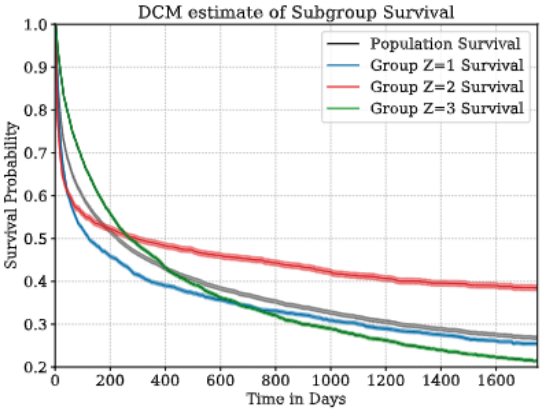
Results



Model	C^{td}		ECE	
	Population	Minority	Population	Minority
CPH	0.6621 ± 0.0087	0.6737 ± 0.0124	0.0992 ± 0.0044	0.0878 ± 0.0071
AFT	0.7911 ± 0.0060	0.7875 ± 0.0087	0.0212 ± 0.0034	0.0329 ± 0.0046
RSF	0.7880 ± 0.0059	0.7830 ± 0.0089	0.0215 ± 0.0037	0.0368 ± 0.0053
FSN	0.6608 ± 0.0081	0.6212 ± 0.0131	0.0381 ± 0.0046	0.0545 ± 0.0068
DHT	0.7636 ± 0.0059	0.7631 ± 0.0092	0.0505 ± 0.0041	0.0525 ± 0.0056
DSM	0.7937 ± 0.0061	0.7909 ± 0.0087	0.0223 ± 0.0029	0.0347 ± 0.0056
DCM	0.7943 ± 0.0103	0.7911 ± 0.0091	0.0200 ± 0.0034	0.0294 ± 0.0049

Results


Kaplan-Meier estimate of Subgroup survival
Population Survival
Group Z=1 Survival
Group Z=2 Survival
Group Z=3 Survival
Survival Probability
Time in Days


DCM estimate of Subgroup Survival
Population Survival
Group Z=1 Survival
Group Z=2 Survival
Group Z=3 Survival
Survival Probability
Time in Days

Recommendation

Worth reading? Yes

- The paper emphasizes the importance of calibration in healthcare settings and highlights the need to address performance disparities in survival analysis models across different demographics, particularly in underrepresented populations.
- The work has clinical relevance and can help healthcare practitioners in risk assessment, triage, and clinical decision-making, with a focus on improving calibration across minority demographics.

Worth implementing? Yes

- Closely related to researches in survival analysis, the installation and users' guidance are shown in github.