Estimation of Conditional Mixture Weibull Distribution with Right Censored Data Using Neural Network for Time-to-Event Analysis

Achraf Bennis(✉), Sandrine Mouysset(✉), and Mathieu Serrurier(✉)

I.R.I.T - Université Toulouse III Paul Sabatier, 31330 Toulouse, France
{achraf.bennis,sandrine.mouysset,mathieu.serrurier}@irit.fr,
bennis.achraf@outlook.fr

Abstract. In this paper, we consider survival analysis with right-censored data which is a common situation in predictive maintenance and health field. We propose a model based on the estimation of two-parameter Weibull distribution conditionally to the features. To achieve this result, we describe a neural network architecture and the associated loss functions that takes into account the right-censored data. We extend the approach to a finite mixture of two-parameter Weibull distributions. We first validate that our model is able to precisely estimate the right parameters of the conditional Weibull distribution on synthetic datasets. In numerical experiments on two real-word datasets (METABRIC and SEER), our model outperforms the state-of-the-art methods. We also demonstrate that our approach can consider any survival time horizon.

Keywords: Survival analysis · Weibull distribution · Neural network

1 Introduction

Time-to-event analysis, also called survival analysis, is needed in many areas. This branch of statistics which emerged in the 20th century is heavily used in engineering, economics and finance, insurance, marketing, health field and many more application areas. Most previous works and diverse literature approach time-to-event analysis by dealing with time until occurrence of an event of interest; e.g. cardiovascular death after some treatment intervention, tumor recurrence, failure of an aircraft air system, etc. The time of the event may nevertheless not be observed within the relevant time period, and could potentially occur after this recorded time, producing so called right-censored data. The main objective of survival analysis is to identify the relationship between the distribution of the time-to-event distribution and the covariates of the observations, such as the features of a given patient, the characteristics of an electronic device

Supported by Smart Occitania, ADEME and ENEDIS.

H. W. Lauw et al. (Eds.): PAKDD 2020, LNAI 12084, pp. 687–698, 2020.
https://doi.org/10.1007/978-3-030-47426-3_53

or a mechanical system with some informations concerning the environment in which it must operate. The Weibull distribution could be used as lifetime distributions in survival analysis where the goal would be to estimate its parameters taking account the right-censored data. Several previous works focused on the estimation of a Weibull distribution with right-censored data (see Bacha and Celeux [1], Ferreira and Silva [2] and Wu [3]).

Among the first estimators widely used in this field is the Kaplan-Meier estimator [4] that may be useful to estimate the probability that an event of interest occurs at a given point in time. However, it is limited in its ability to estimate this probability adjusted for covariates; i.e. it doesn't incorporate observations' covariates. The semi-parametric Cox proportional hazards (CPH) [5] is used to estimate covariate-adjusted survival, but it assumes that the subject's risk is a linear function of their covariates which may be too simplistic for many real world data. Since neural networks can learn nonlinear functions, many researchers tried to model the relationship between the covariates and the times that passes before some event occurs, including Faraggi-Simon network [6] who proposed a simple feed-forward as the basis for a non-linear proportional hazards model to model this relationship. After that, several works focused on combining neural networks and survival analysis, notably DeepSurv [7] whose architecture is deeper than Faraggi-Simon's one and minimizes the negative log Cox partial likelihood with a risk not necessarily linear. These models use multilayer perceptron that is capable to learn non-linear models, but it is sensitive to feature scaling which is necessary in data preprocessing step and has limitations when we use unstructured data (e.g. images). There is a number of other models that approach survival analysis with right-censored data using machine learning, namely RandomForest Survival [8], dependent logistic regressors [9] and Liao's model [10] who are capable of incorporating the individual observation's covariates. This paper proposes a novel approach to survival analysis: we assume that the survival times distribution are modeled according to a finite Mixture of Weibull distributions (at least one), whose parameters depends on the covariates of a given observations with right-censored data. As Luck [11], we propose a deep learning model that learns the survival function, but we will do this by estimating the Weibull's parameters. Unlike DeepHit [12] whose model consists on discretizing the time considering a predefined maximum time horizon. Here, as we try to estimate the parameters, we can model a continuous survival function, and thus, estimate the risk at any given survival time horizon. For this purpose, we construct a deep neural network model considering that the survival times follow a finite mixture of two-parameter Weibull distributions. This model, which we call **DeepWeiSurv** tries to estimate the parameters that maximize the likelihood of the distribution. To prove the usefulness of our method, we compare its predictive performance with that of state-of-the art methods using two real-world datasets. DeepWeiSurv outperforms the previous state-of-the-art methods.

2 Weibull Mixture Distribution for Survival Analysis

2.1 Survival Analysis with Right-Censored Data

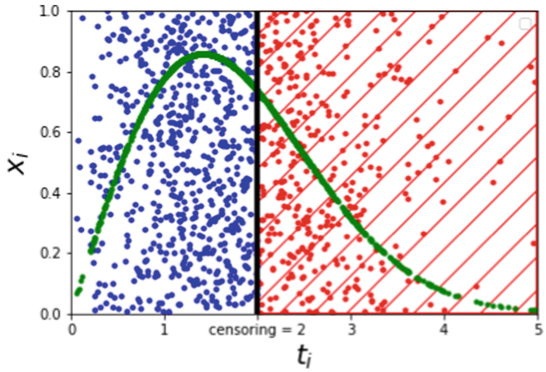

Fig. 1. Weibull distribution right-censored at $t_c = 2$ with $x \in [0, 1]$ uniformly distributed. In this figure, the parameters of the law are independent with regard to x. (Color figure online)

Let $X = \{(x_i, t_i, \delta_i)|i \leq n\}$ be a set of observations with $x_i \in \mathbb{R}^d$, the i^{th} observation of the baseline data (covariates), $t_i \in \mathbb{R}$ its survival time associated, and δ_i indicates if the i^{th} observation is censored ($\delta_i = 0$) or not ($\delta_i = 1$). As can be seen in Fig. 1, a blue point represents an uncensored observation $(x_i, t_i, \delta_i = 1)$ and a red point represents a censored observation $(x_i, t_i, \delta_i = 0)$. In order to characterize the distribution of the survival times $T = (t_i|x_i)_{i \leq n}$, the aim is to estimate, for each observation, the probability that the event occurs after or at a certain survival time horizon t_{STH} defined by:

$$S(t_i|x_i) = P(t_i \geq t_{STH}|x_i). \tag{1}$$

Note that, t_{STH} may be different to the censoring threshold time t_c. An alternative characterization of the distribution of T is given by the hazard function $\lambda(t)$ that is defined as the event rate at time t conditional on survival at time t or beyond. Literature has shown that $\lambda(t)$ can be expressed as follows: $\lambda(t) = \frac{f(t)}{S(t)}$, $f(t)$ being the density function.

Instead of estimating the $S(t_i|x_i)$, it is common to estimate directly the survival time $\hat{t}_i$. In this case, we can measure the quality of estimations with the concordance index [13] defined as follows:

$$C_{index} = \frac{\sum_{i,j} \mathbb{1}_{t_i > t_j} . \mathbb{1}_{\hat{t}_i > \hat{t}_j} . \delta_j}{\sum_{i,j} \mathbb{1}_{t_i > t_j} . \delta_j}. \tag{2}$$

C_{index} is designed to calculate the number of concordant pairs of observations among all the comparable pairs (i, j) such that $\delta_i = \delta_j = 1$. It estimates the probability $\mathbb{P}(\hat{t}_i > \hat{t}_j | t_i > t_j)$ that compares the rankings of two independent pairs of survival times t_i, t_j and associated predictions $\hat{t}_i$, $\hat{t}_j$.

2.2 Weibull Distribution for Censored Data

From now, we consider that T follows a finite mixture of two-parameter Weibull (at least a single Weibull) distributions independently from x_i (i.e. $S(t_i|x_i) = S(t_i)$). In this case, we have the analytical expressions of S and λ with respect to the mixture parameters. This leads to consider a problem of parameters estimation of mixture of Weibull distributions with right-censored observations. Let $y = (y_i)_i = (t_i, \delta_i)_i$.

Single Weibull Case. Here, we are dealing with a particular case where T follows a single two-parameter Weibull distribution, $\mathscr{W}(\beta, \eta)$, whose parameters are $\beta > 0$ (*shape*) and $\eta > 0$ (*scale*). We can estimate these parameters by solving the following likelihood optimization problem:

$$\hat{\beta}, \hat{\eta} = \underset{\beta,\eta}{\text{argmax}}\{\mathcal{LL}(\beta, \eta|y\} = \sum_{i=1}^{n} \delta_i log[(S_{\beta,\eta}.\lambda_{\beta,\eta})(t_i)] + (1 - \delta_i) log[S_{\beta,\eta}(t_c)] \quad (3)$$

where:

$$\begin{aligned} S_{\beta,\eta}(t) &= exp[-(\frac{t}{\eta})^{\beta}], \\ \lambda_{\beta,\eta}(t) &= (\frac{\beta}{\eta})(\frac{t}{\eta})^{\beta-1} \end{aligned} \quad (4)$$

and t_c being the censoring threshold time. $\mathcal{LL}$ is the log-likelihood of Weibull distribution with right-censored data. To be sure that the $\mathcal{LL}$ is concave, we make a choice to consider that the shape parameter β is greater than 1 ($\beta \geq 1$).

Mixture Case. Now, we suppose that T follows $\mathcal{W}_p = [(\mathscr{W}(\beta_k, \eta_k)), (\alpha_k)]_{k=1..p}$ a mixture of p Weibull distributions with its weighting coefficients ($\sum_k \alpha_k = 1$, $\alpha_k \geq 0$). In statistics, the density associated is defined by:

$$f_{\mathcal{W}_p} = \sum_k \alpha_k f_{\beta_k,\eta_k} = \sum_k \alpha_k S_{\beta_k,\eta_k} \lambda_{\beta_k,\eta_k}. \quad (5)$$

Thus, the log-likelihood of $\mathcal{W}_p$ can be written as follows:

$$\begin{aligned} \mathcal{LL}(\beta, \eta, \alpha|y) = \sum_{i=1}^{n} & \delta_i log\left[\textstyle\sum_k \alpha_k (S_{\beta_k,\eta_k}.\lambda_{\beta_k,\eta_k})(t_i)\right] \\ & + (1 - \delta_i) log\left[\textstyle\sum_k \alpha_k S_{\beta_k,\eta_k}(t_c)\right]. \end{aligned} \quad (6)$$

In addition to the mixture's parameters $(\beta_k, \eta_k)_{k=1...p}$, we need to estimate the weighting coefficients (α_k) considered as probabilities. Therefore, we estimate the tuple (α, β, η) by solving the following problem:

$$(\hat{\beta}, \hat{\eta}, \hat{\alpha}) = \underset{\beta,\eta,\alpha}{\operatorname{argmax}}\{\mathcal{LL}(\beta, \eta, \alpha|(t_i, \delta_i)_i)\} \tag{7}$$

Knowing Weibull's mean formula μ and given that the mean of a mixture is a weighted combination of the means of the distributions that form this mixture (more precisely, $\mu = \sum_k \alpha_k \mu_k$), the mean lifetime can thus be estimated as follows:

$$\mu = \hat{\alpha}.diag(\hat{\eta}).\mathit{\Gamma}(1 + \frac{1}{\hat{\beta}})^T \tag{8}$$

where $\mathit{\Gamma}$ is the Gamma function. μ can be used as the survival time estimation for the computation of the concordance index (with $\hat{t_i} = \mu_i = \mu$ when the parameters of the distribution are independent from x_i).

3 Neural Network for Estimating Conditional Weibull Mixture

We now consider that the Weibull mixture's parameters depend on the covariates $x = (x_i)$. We propose to use a neural network to model this dependence.

3.1 Model Description

We name g_p the function that models the relationship between x_i and the parameters of the conditional Weibull mixture:

$$\begin{aligned} g_p : \mathbb{R}^d &\rightarrow \quad \mathbb{R}^{p\times 3} \\ x_i &\mapsto (\alpha, \beta, \eta) \end{aligned} \tag{9}$$

where $\alpha = (\alpha_1, ..\alpha_p)$ and $(\beta, \eta) = ((\beta_1, .., \beta_p), (\eta_1, .., \eta_p))$. Note that, when $p = 1$, it is no more required to estimate α. This function is represented by the network named DeepWeiSurv described in Fig. 2. Hence, our goal is to train the network to learn g_p and thus $(\hat{\beta}, \hat{\eta})$ the vector of parameters that maximize the likelihood of the time-to-event distribution ($\hat{\alpha}$ as well if $p > 1$). DeepWeiSurv is therefore a multi-task network. It consists of a common sub-network, a classification sub-network (*clf*) and a regression sub-network (*reg*). The shared sub-network takes as an input the baseline data $\mathbf{x}$ of size n and compute a latent representation of the data z. When $p > 1$, *clf* and *reg* take z as an input towards producing $\hat{\alpha}$ and $(\hat{\beta}, \hat{\eta})$ respectively. For *reg* sub-network, we use ELU (with its constant = 1) as an activation function for both output layers. We use this function to be sure that we have enough gradient to learn the parameters thanks to the fact that it becomes smooth slowly unlike ReLU function. However the codomain of ELU is $]-1, \infty[$, which is problematic given the constraints on the parameters mentioned in the previous section ($\beta \geq 1$ and $\eta > 0$). To get around this problem, the network

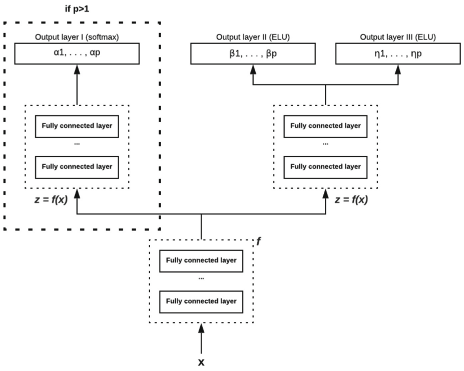

Fig. 2. The architecture of DeepWeiSurv

will learn $\hat{\beta}_{off} = \hat{\beta} + 2$ and $\hat{\eta}_{off} = \hat{\eta} + 1 + \epsilon$. The offset is then applied in the opposite direction to recover the parameters concerned. For the classification part we need to learn $\alpha \in \mathbb{R}^p$. To ensure that $\sum_k \alpha_k = 1$ and $\alpha_k \in [0,1]$, we use a *softmax* activation in the output layer of *clf*. For each $1 \leq k \leq p$, *clf* produces, $\alpha_k = (\alpha_{1k}, ...\alpha_{nk})$ where α_{ik} is such that: $\hat{P}(\{Y = t_i\}) = \alpha_{ik}$ with $Y \sim \mathscr{W}(\beta_k, \eta_k)$ and $\hat{P}$ a probability estimate, whereas *reg* outputs $\beta_k = (\beta_{1k}, ..\beta_{nk})$ and $\eta_k = (\eta_{1k}, ...\eta_{nk})$. Otherwise, i.e. $p = 1$, we have $\alpha_1 = 1$, thus we don't need to train *clf*.

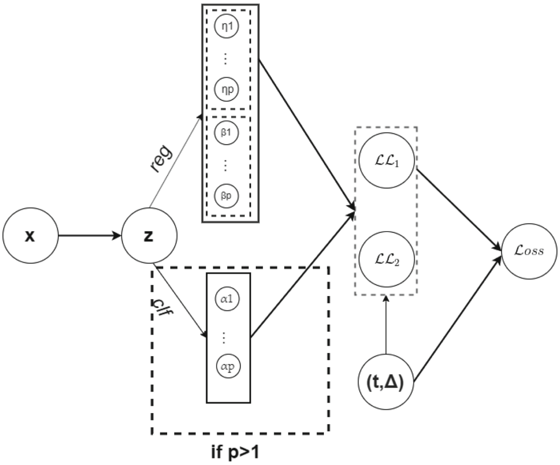

Fig. 3. Computational graph of $\mathcal{L}oss$

To train DeepWeiSurv, we minimize the following loss function:

$$\mathcal{L}oss = -\mathcal{LL}(\beta, \eta, \alpha | y) \underset{(6)}{=} \mathcal{LL}_1.\Delta^T + \mathcal{LL}_2.(\mathbf{1}_{\mathbb{R}^n} - \Delta)^T \qquad (10)$$

where Δ is the vector of event indicators and:

$$\mathcal{LL}_1 = log[\hat{\alpha}.S\Lambda_{\hat{\beta},\hat{\eta}}(T)] \text{ and } \mathcal{LL}_2 = log[\hat{\alpha}.S_{\hat{\beta},\hat{\eta}}(t_c)] \tag{11}$$

with:

$$S\Lambda_{\hat{\beta},\hat{\eta}}(t) = \begin{pmatrix} (S_{\beta_1,\eta_1}.\lambda_{\beta_1,\eta_1})(t_1) & ... & (S_{\beta_1,\eta_1}.\lambda_{\beta_1,\eta_1})(t_n) \\ ... & ... & ... \\ (S_{\beta_p,\eta_p}.\lambda_{\beta_p,\eta_p})(t_1) & ... & (S_{\beta_p,\eta_p}.\lambda_{\beta_p,\eta_p})(t_n) \end{pmatrix} \tag{12}$$

and

$$S_{\hat{\beta},\hat{\eta}}(t_c) = \begin{pmatrix} S_{\beta_1,\eta_1}(t_c) \\ ... \\ S_{\beta_p,\eta_p}(t_c) \end{pmatrix} \tag{13}$$

$\mathcal{LL}_1$ exploits uncensored data, whereas $\mathcal{LL}_2$ exploits censored observations by extracting the knowledge that the event will occur after the given censoring threshold time t_c. Figure 3 is an illustration of the computational graph of our training loss: the inputs are the covariates x, the real values of time and event indicator (t, Δ) and the outputs are the estimates $(\hat{\alpha}, \hat{\beta}, \hat{\eta})$.

3.2 Experiment on SYNTHETIC Dataset

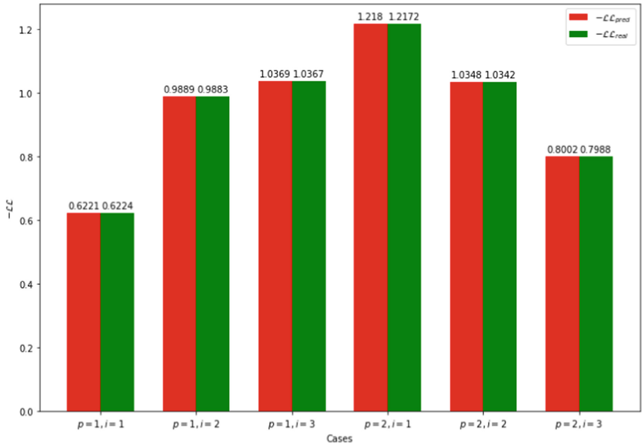

Fig. 4. $-\mathcal{LL}_{pred}$ and $-\mathcal{LL}_{real}$ values for each studied case

The main objective in this section is to validate mathematically DeepWeiSurv, that is, to show that this latter is able to estimate the parameters. For this purpose, we perform an experiment on a simulated data. In this experiment, we treat the case of a *single* Weibull distribution ($\alpha_{p=1} = 1$) and a *mixture* of 2 Weibull distributions ($\alpha_{p=2} = (0.7, 0.3)$) using three different functions: f_1 (*linear*), f_2 (*quadratic*), f_3 (*cubic*). For each function f_i we generate $T^i_{p=1}$

$\sim \mathscr{W}(\beta_1^i, \eta_1^i)$ and $T^i_{p=2} \sim \mathcal{W}_{p=2}(\beta^i_{0.7}, \eta^i_{0.7}, \beta^i_{0.3}, \eta^i_{0.3})$. We compare the predicted likelihood with the real, and optimal one. These two likelihoods are equal when the estimated parameters correspond to the real ones. Let X be a vector of 10000 observations generated from an uniform distribution $\mathcal{U}_{[0,1]}$. Here we select 50% of observations to be right censored at the median of survival times t_m ($\delta_i = 0$ if $t_i > t_m$). We set the parameters to be the following functions:

$$\begin{pmatrix} \beta_1^1 \\ \eta_1^1 \\ \beta_{0.7}^1 \\ \eta_{0.7}^1 \\ \beta_{0.3}^1 \\ \eta_{0.3}^1 \end{pmatrix} = \begin{pmatrix} 3 & 2 \\ 2 & 1 \\ 2 & 1 \\ 1 & 2 \\ 1 & 2 \\ 3 & 1 \end{pmatrix} \cdot \begin{pmatrix} X \\ 1 \end{pmatrix} \begin{pmatrix} \beta_1^2 \\ \eta_1^2 \\ \beta_{0.7}^2 \\ \eta_{0.7}^2 \\ \beta_{0.3}^2 \\ \eta_{0.3}^2 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 2 & 2 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 2 \\ 1 & 0 & 2 \end{pmatrix} \cdot \begin{pmatrix} X^2 \\ X \\ 1 \end{pmatrix} \begin{pmatrix} \beta_1^3 \\ \eta_1^3 \\ \beta_{0.7}^3 \\ \eta_{0.7}^3 \\ \beta_{0.3}^3 \\ \eta_{0.3}^3 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 2 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 2 & 0 & 1 \\ 3 & 2 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} X^3 \\ X^2 \\ X \\ 1 \end{pmatrix}$$

The bar plot in Fig. 4 displays the predicted likelihood $-\mathcal{LL}_{pred}$ of each distribution and their real one $-\mathcal{LL}_{real}$. We notice that the real value and predicted one of each case are very close to each other which means that the model can identify very precisely the parameters of the conditional distributions. Now, we test DeepWeisurv on the real-world datasets.

4 Experiments

We perform two sets of experiments based on real survival data: METABRIC and SEER. We give a brief descriptions of the datasets below; Table 1 gives an overview on some descriptive statistics of both real-word datasets. We train DeepWeiSurv on real survival datasets. We compare the predictive performance of DeepWeiSurv with that of CPH [5] which is the most-widely used model in the medical field and DeepHit [12] that seems to achieve outperformance over previous methods. These models are also tested in the same experimental protocol as DeepWeiSurv.

METABLNCSRIC. METABRIC (Molecular Taxonomy of Breast Cancer International Consortium) dataset is for a Canada-UK project that aims to classify breast tumours into further subcategories. It contains gene expressions profiles and clinical features used for this purpose. In this data, we have 1981 patients, of which 44.8% were died during the study and 55.2% were right-censored. We used 21 clinical variables including tumor size, age at diagnosis, Progesterone Receptor (PR) status etc. (see Bilal et al. [14]).

SEER. The Surveillance, Epidemiology, and End Results (SEER[1]) [15] Program provides information on cancer statistics during 1975–2016. We focused on the patients (in total 33387) recorded between 1998 and 2002 who died from a breast cancer BC (42.8%) or a heart disease HD (49.6%), or who were right-censored

[1] https://seer.cancer.gov.

(57.2% and 50.4% respectively). We extracted 30 covariates including gender, race, tumor size, number of malignant of benign tumors, Estrogen Receptor status (ER), PR status, etc. For evaluation we separated the data into two datasets with respect of the death's cause (BC & HD) while keeping censored patients in both of them.

Table 1. Descriptive statistics of real-world datasets

Datasets	No. uncensored	No. censored	No. features	
			Qualitative	Quantitative
METABRIC	888 (44.8%)	1093 (55.2%)	15	6
SEER BC	9152(42.8%)	12221 (57.2%)	23	11
SEER HD	12014 (49.6%)	12221 (50.4%)		

4.1 Network Configuration

DeepWeiSurv is consisted of three blocks: the shared sub-network which is a 4-layer network, 3 of which are fully connected layers (128, 64, 32 nodes respectively) and the remain is a batch normalization layer, the second and the third block (*reg*, *clf* respectively) consisted of 2 fully connected layers (16, 8 nodes) and 1 batch normalization layer. Added to that, the network finishes by one softmax layer and two ELU layers as outputs. The hidden layers are activated by ReLU function. DeepWeiSurv is trained via Adam optimizer and learning rate of 10^{-4}. DeepWeiSurv is implemented in a PyTorch environment.

4.2 Experimental Protocol

We applied 5-fold cross validation: the data is randomly splitted into training set (80% and 20% of which is reserved for validation) and test set (20%). We use the predicted values of the parameters to calculate the mean lifetime μ and then C_{index} defined by Eq. (2). This latter is calculated on the validation set. We tested DeepWeiSurv with $p = 1$ and $p = 2$ (we tested higher values of p, but without better performance).

4.3 Results

Table 2 displays the C_{index} results of the experiments realized on SEER and METABRIC datasets. We can observe that, for METABRIC, DeepWeiSurv's performances exceed by far that of DeepHit and CPH. For the SEER data, DeepWeiSurv with $p = 1$ outperfoms CPH (in BC and HD cases) and has a slight improvement over DeepHit especially for SEER HD data but without a significant difference (their confidence intervals did overlap). However, the improvement

Table 2. Comparison of C_{index} performance tested on METABRIC and SEER (mean and 95% confidence interval)

Algorithms	Datasets		
	METABRIC	SEER BC	SEER HD
CPH	0.658 (0.646–0.671)	0.833 (0.829–0.838)	0.784 (0.779–0.788)
DeepHit	0.651 (0.641–0.661)	0.875 (0.867–0.883)	0.846 (0.842–0.851)
DeepWeiSurv ($p = 1$)	0.805 (0.782–0.829)	0.877 (0.864–0.891)	0.857 (0.85–0.866)
DeepWeiSurv ($p = 2$)	**0.819 (0.812–0.837)**	**0.908 (0.906–0.909)**	**0.863 (0.86–0.868)**

of DeepWeiSurv with $p = 2$ over all the other methods is highly statistically significant. We suspect that the good performances of DeepWeiSurv comes from its ability to learn implicitly the relationship between the covariates and the parameters without making any restrictive assumption.

4.4 Censoring Threshold Sensitivity

In the previous experiments the survival time horizon and the censoring threshold coincide, but it is not always the case. Since DeepWeiSurv predicts the conditional Weibull distributions with respect to the covariates, it is able to consider any survival time horizon given a censoring threshold. We add another experiment on METABRIC[2] dataset where we assess DeepWeiSurv ($p = 2$) performance with respect to censoring threshold time t_c. The aim of this experiment, is to check if DeepWeiSurv can handle data in highly censored setting for different survival time horizons. For this purpose, we apply the same experimental protocol as before, but changing the censoring threshold. We do this for some values of t_c far below than that used in the previous experiment ($t_c = t_{METABRIC} = 8940$). This values, expressed in quantiles[3], are carefully selected in order to have a significant added portion (compared to that of the adjacent value that precedes) of censored observations. As an observation may change from a censored status to an uncensored status by changing the threshold of censorship and vice versa, for each value of censoring threshold time t_c we therefore have a new set of observed events $OE_{t_c} = \{(t_i, \delta_i) \| \delta_i = 1 \text{ if } t_i < t_c \; else \; 0\}$ (i.e. comparable events, and this contributes to the calculation of C_{index}). The training set, as it is selected, contains $ref = 866$ censored observations. Table 3 gives the number of censored and uncensored observations of each selected value of t_c. For each value of t_c, we apply the 5-fold cross validation and then calculate the average C_{index} for every survival time horizons t_{STH}. The results are displayed in Fig. 5.

[2] We have chosen METABRIC dataset because of its small size compared to that of SEER dataset in order to avoid long calculations.

[3] We choose this values by using the quantiles of the survival times vector T.

Table 3. Distribution of training set's observations (censored/uncensored) for each selected censoring threshold.

t_c	No. censored	No. uncensored	Added portion (w.r.t ref)
$q_{0.5}$	1026	558	160
$q_{0.45}$	1127	457	261
$q_{0.35}$	1248	336	382
$q_{0.25}$	1338	246	472

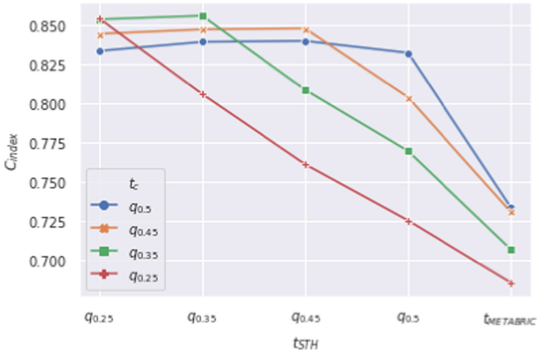

Fig. 5. The average of C_{index} w.r.t survival time horizon t_{STH} for every selected threshold t_c.

Each curve in Fig. 5 represents the scores calculated for a given censoring threshold t_c in different survival time horizons t_{STH} in x-axis. We can notice that the average score decreases when t_c decreases which is expected because we have less and less of uncensored data which means that it becomes more and more difficult to model the distribution of survival times. However, DeepWeiSurv still performing well in highly censored setting.

5 Conclusion

In this paper, we described a new approach, DeepWeiSurv, to the survival analysis. The key role of DeepWeiSurv is to predict the parameters of a mixture of Weibull distributions with respect to the covariates in presence of right-censored data. In addition to the fact that Weibull distributions are known to be a good representation for this kind of problem, it also permits to consider any survival time horizon given a censoring threshold. Experiments on generated databases show that DeepWeiSurv converges to the real parameters when the survival time data follows a mixture of Weibull distributions whose parameters are a simple function of the covariates. On real datasets, DeepWeiSurv clearly outperforms the state-of-the-art approaches and demonstrates its ability to consider any survival time horizon.

References

1. Bacha, M., Celeux, G.: Bayesian estimation of a Weibull distribution in a highly censored and small sample setting. INRIA (1996)
2. Ferreira, L.A., Silva, J.L.: Parameter estimation for Weibull distribution with right censored data using EM algorithm. Eksploatacja i Niezawodność **19**, 310–315 (2017)
3. Wu, S.-J.: Estimations of the parameters of the Weibull distribution with progressively censored data. J. Jpn. Stat. Soc. **32**, 155–163 (2002)
4. Marin, J.-M., Mengersen, K., Robert, C.P.: Bayesian modelling and inference on mixtures of distributions. In: Handbook of Statistics, vol. 25, pp. 459–507. Elsevier (2005)
5. Cox, D.R.: Regression models and life tables (with discussion). J. Roy. Stat. Soc. Ser. B. **34**, 187–220 (1972)
6. Faraggi, D., Simon, R.: A neural network model for survival data. Stat. Med. **14**, 73–82 (1995)
7. Katzman, J.L., Shaham, U., Cloninger, A., Bates, J., Jiang, T., Kluger, Y.: Deep survival: a deep Cox proportional hazards network. Statistics **1050**, 2 (2016)
8. Ishwaran, H., Kogalur, U.B., Blackstone, E.H., Lauer, M.S., et al.: Random survival forests. Ann. Appl. Stat. **2**, 841–860 (2008)
9. Yu, C.-N., Greiner, R., Lin, H.-C., Baracos, V.: Learning patient-specific cancer survival distributions as a sequence of dependent regressors. In: Advances in Neural Information Processing Systems 24, pp. 1845–1853. Curran Associates Inc. (2011)
10. Liao, L., Ahn, H.-I.: Combining deep learning and survival analysis for asset health management. Int. J. Progn. Health Manag. (2016)
11. Luck, M., Sylvain, T., Cardinal, H., Lodi, A., Bengio, Y.: Deep learning for patient-specific kidney graft survival analysis. arXiv preprint arXiv:1705.10245 (2017)
12. Lee, C., Zame, W.R., Yoon, J., van der Schaar, M.: DeepHit: a deep learning approach to survival analysis with competing risks. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
13. Harrell, F.E., Califf, R.M., Pryor, D.B., Lee, K.L., Rosati, R.A.: Evaluating the yield of medical tests. JAMA **247**(18), 2543–2546 (1982)
14. Bilal, E., et al.: Improving breast cancer survival analysis through competition-based multidimensional modeling. PLoS Comput. Biol. **9**, e1003047 (2013)
15. National Cancer Institute, DCCPS, Surveillance Research Program: Surveillance, Epidemiology, and End Results (SEER) Program. www.seer.cancer.gov. SEER*Stat Database: Incidence - SEER 18 Regs Research Data + Hurricane Katrina Impacted Louisiana Cases, November 2018 Submission (1975–2016 varying). Linked To County Attributes - Total U.S., 1969–2017 Counties, Released April 2019, Based on the November 2018 Submission