Neural Network for Correlated Survival Outcomes Using Frailty Model

Anonymous Authors[1]

Abstract

Extensive literature has been proposed for the analysis of correlated survival data. Subjects within a cluster share some common characteristics, e.g., genetic and environmental factors, so their time-to-event outcomes are correlated. The frailty model under proportional hazards assumption has been widely applied for the analysis of clustered survival outcomes. However, the prediction performance of this method can be less satisfactory when the risk factors have complicated effects, e.g., nonlinear and interactive. To deal with this issue, we propose a neural network frailty Cox model that replaces the linear risk function with the output of a feed-forward neural network. The estimation is based on quasi-likelihood with the use of Laplace approximation. A simulation study suggests that the proposed method has the best performance compared with five existing methods. The method is applied to the clustered time-to-glaucoma in both eyes from the Ocular Hypertension Treatment Study (OHTS) study.

1. Introduction

Survival models have been extensively developed in medical research to make inferences and predictions on failure times. The Cox proportional hazards model is the most commonly used regression model for survival outcomes. In the conventional Cox model, the survival outcomes from different observational units are assumed to be independent, given the current time and observed covariates. However, survival outcomes may be correlated. In our motivating example of the Ocular Hypertension Treatment Study (OHTS, Kass et al. 2002, 2010, 2021), the incidence of primary open-angle glaucoma (POAG) was recorded for both eyes. The hazards of POAG on the two eyes from the same patient share specific unobserved characteristics and therefore tend to be correlated. Ignoring such associations will lead to inefficient and biased estimation for the prediction of the time-to-event.

To account for within-cluster dependency, extensive literature has been published on frailty models, where the survival outcomes are assumed to be independent conditional on unobserved frailty (random effect) terms. In the Cox proportional hazards frailty model, the frailty or random effect is assumed to follow a probability distribution (Balan and Putter 2020). For example, Paik, Tsai, and Ottman (1994), Shih and Louis (1995), and Hens et al. (2009) assumed the frailty follows a gamma distribution, while Ripatti and Palmgren (2000) considered the log-normal frailty distribution. Other examples include the power-variance-function (PVF) family, where the marginal distribution of survival outcome can be obtained in a closed form. Besides the frailty models, the stratified Cox model is a popular tool for clustered survival outcome because of its simplicity in computation and interpretation. However, according to Gidden and Vittinghoff (2004), the stratified Cox model discards between-cluster comparison information, leading to inefficient estimation and a loss of efficiency, particularly when there are a large number of strata or clusters as in the paired survival outcomes (e.g., both eyes) in the OHTS study.

Recently, deep learning methods for time-to-event prediction have gained significant attention and success. Deep learning methods in survival models are shown to have better predictive power than traditional survival models, especially with the existence of highly nonlinear and interactive risk effects. Liao et al. (2016), Martinsson (2016), and Ranganath et al. (2016) proposed deep learning algorithms when the survival outcomes were assumed to follow the Weibull distribution. Under the semi-parametric Cox proportional hazards model framework, Faraggi and Simon (1995) first adopted a feed-forward neural network. Later, Katzman et al. (2018) proposed an algorithm "Deepsurv" by minimizing the loss function derived from the partial likelihood function and utilizing modern deep learning techniques. Ching, Zhu, and Garmire (2018) suggested Cox-nnet for high-throughput RNA sequencing data. Hao et al. (2018) illustrated Cox-PASNet method, which integrates high-dimensional gene expression data and clinical data on a simple neural network architecture for survival analysis to improve the biological interpretation of genes and pathways.

Although many deep learning methods have been investi-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

gated for survival outcomes, few have tackled correlated survival outcomes. In this paper we will consider deep learning methods for correlated survival outcomes, e.g., paired eyes, mice in the same litter. It should be noted that the clusters of interest are pre-defined and cluster memberships are fully observed (e.g., eyes from the same subject, mice from the same litter). It differs from clustering analysis methods in the machine learning or deep learning content, in which the goal is to classify subjects according to their underlying characteristics.

We propose a neural network for predicting correlated survival time under the Cox proportional hazards frailty model. The frailty term is assumed to follow a normal distribution. We predict the risk score as a nonlinear function of covariates with a feed-forward neural network. We maximize the penalized partial likelihood with Laplace approximation to tackle the computational difficulties and formulate the loss function with the penalized partial likelihood. Simulation studies are conducted to compare the predictive accuracy of the proposed method relative to competing methods. In addition, we apply the proposed method to investigate the time to POAG for both eyes on the same subject in the OHTS study.

The rest of the paper is organized as follows. Section 2 describes the proposed deep-learning method for correlated survival outcomes. In Section 3, we assess the performance of our method via simulation studies. Section 4 illustrates the proposed method by applying it to the OHTS and compares the prediction performance with competing methods. We summarize our method and present some future directions in Section 5.

2. Model Specification

2.1. Problem Formulation

Let T_{ij} denote the event time for the j-th unit within i-th cluster, where $i = 1, \ldots, s$ and $j = 1, \ldots, c$. The sample size $n = s \times c$. We denote C_{ij} as the censoring time, $U_{ij} = \min(T_{ij}, C_{ij})$ as the observed time, and $\Delta_{ij} = I_{\{T_{ij} \leq C_{ij}\}}$ as the right-censored indicator. Given random effect, or frailty $\boldsymbol{b}_i$, the event times are assumed independent with the conditional hazard function

$$\lambda_{ij}(t \mid b_i) = \lambda_0(t) \exp\left(\boldsymbol{X}_{ij}^T \boldsymbol{\beta} + \boldsymbol{Z}_{ij}^T \boldsymbol{b}_i\right),$$

where $\boldsymbol{X}_{ij}$ and $\boldsymbol{Z}_{ij}$ are vectors of explanatory variables; $\lambda_0(t)$ is the baseline hazards function; and $\boldsymbol{b}_i$ is the vector of frailties (random effects). In the most commonly used frailty model, $Z_{ij} = 1$, i.e., only the random intercept is of interest. Hence, we will simplify our illustration by only considering random intercept b_i, which is assumed to follow a normal distribution with mean 0 and variance θ.

To better describe the covariate effects, we consider a feed-forward artificial neural network (FNN) with L hidden layers. We adapt the classical FNN under Cox proportional hazards model to a deep learning method within the frailty model framework, which may lead to more precise hazard function estimates and improved survival predictions. The covariate $\boldsymbol{X}_{ij}$ are p inputs or p variables, and $\boldsymbol{X}_{ij}^T \boldsymbol{\beta}$ can be replaced by a nonlinear function of the predictors $\boldsymbol{X}_{ij}$ with network weights $\boldsymbol{\omega}^{(l)}$ and bias $\boldsymbol{\delta}^{(l)}$ through a series of nested activation function $g_l(\cdot)$ for layers $l = 0, \ldots, L$. Weights and biases are also called slope coefficients and intercepts, respectively, in statistical terms. To be specific, the k_0 nodes of the first hidden layer can be calculated through

$$\boldsymbol{\alpha}_{ij}^{(0)} = g_0\left\{\boldsymbol{\omega}^{(0)} \boldsymbol{X}_{ij} + \boldsymbol{\delta}^{(0)}\right\},$$

where $\boldsymbol{\omega}^{(0)}$ is a $k_0 \times p$ weight matrix, $\boldsymbol{\delta}^{(0)}$ is a bias vector of length k_0, and the activation function $g_0(\cdot)$ is applied element-wise to its input vector. For the l th hidden layer $(l = 1, \ldots, L-1)$ with k_l nodes, the layer's output is

$$\boldsymbol{\alpha}_{ij}^{(l)} = g_l\left\{\boldsymbol{\omega}^{(l)} \boldsymbol{\alpha}_{ij}^{(l-1)} + \boldsymbol{\delta}^{(l)}\right\},$$

where $\boldsymbol{\omega}^{(l)}$ is a $k_l \times k_{l-1}$ matrix and $\boldsymbol{\delta}^{(l)}$ is of length k_l. Finally, when only random intercept is considered, the univariate output from the neural network is related to the proportional hazards function by

$$\lambda_{ij}^{NN}(t \mid b_i) = \lambda_0(t) \exp(\alpha_{ij}^{(L)} + b_i), \tag{1}$$

where $\alpha_{ij}^{(L)} = g_L(\boldsymbol{\omega}^{(L)} \boldsymbol{\alpha}_{ij}^{(L-1)})$, $\boldsymbol{\alpha}_{ij}^{(L-1)}$ is the second last layer's output , and $\boldsymbol{\omega}^{(L)}$ is a $1 \times k_{L-1}$ vector.

2.2. Penalized Partial Likelihood

Conditionally on b_i, the likelihood for cluster i in model (1) is

$$L_i^{NN}\left(\lambda_0(t), \boldsymbol{\omega}, \boldsymbol{\delta}, \theta | b_i\right) = \int \prod_{j=1}^{c} \exp\left[l_{ij}^{NN}\left(\lambda_0(t), \boldsymbol{\omega}, \boldsymbol{\delta} | b_i\right)\right] p(b_i; \theta) db_i,$$

where

$$l_{ij}^{NN}\left(\lambda_0(t), \boldsymbol{\omega}, \boldsymbol{\delta} | b_i\right) = \Delta_{ij}\left[\log\left(\lambda_0(t)\right) + \alpha_{ij}^{(L)} + b_i\right] - \Lambda_0(t) \exp(\alpha_{ij}^{(L)} + b_i)$$

and

$$p(b_i; \theta) = \theta^{-1/2}(2\pi)^{-1/2} exp(-\frac{1}{2} b_i{}' \theta^{-1} b_i).$$

The function $\Lambda_0(t) = \int_0^t \lambda_0(u) du$ is the baseline cumulative hazards function, $l_{ij}^{NN}(\cdot | b_i)$ denotes the log likelihood function for subject j in the i-th cluster given random

effect b_i, and the unobserved frailties are integrated out. The parameter $\boldsymbol{\omega}$ represents the combined vectorization of $\boldsymbol{\omega}^{(0)}, \ldots, \boldsymbol{\omega}^{(L)}$ into a single column vector, $\boldsymbol{\delta}$ represents the concatenation of $\boldsymbol{\delta}^{(0)}, \ldots, \boldsymbol{\delta}^{(L-1)}$ into a column vector. To avoid overfitting, we add some penalization to the neural network parameter $\boldsymbol{\omega}$ and $\boldsymbol{\delta}$, with regularization parameter γ.

The likelihood function for model (1) for cluster i with parameter regularization then becomes

$$\tilde{L}_i^{NN}(\lambda_0(t), \boldsymbol{\omega}, \boldsymbol{\delta}, \theta) = \theta^{-1/2}(2\pi)^{-1/2} \int \exp\Bigg[\sum_{i=1}^{c}\{ \Delta_{ij}\Big[\log\big(\lambda_0(t)\big) + \alpha_{ij}^{(L)} + b_i\Big] - \Lambda_0(t)\times \exp(\alpha_{ij}^{(L)} + b_i) - \frac{1}{2} b_i{}' \theta^{-1} b_i - \boldsymbol{\gamma}(\boldsymbol{\omega}^T\boldsymbol{\omega} + \boldsymbol{\delta}^T\boldsymbol{\delta})\}\Bigg] db_i. \quad (2)$$

Under the normal distribution assumption for the frailty term, equation (2) is difficult to maximize with an integral. Following Ripatti and Palmgren (2000), we use a Laplace approximation for the integral in $\tilde{L}_i^{NN}\big(\lambda_0(t), \boldsymbol{\omega}, \boldsymbol{\delta}, \theta\big)$. This leads to the approximated marginal log-likelihood for cluster i,

$$\log(\tilde{L}_i^{NN}) = l_i\big(\lambda_0(t), \boldsymbol{\omega}, \boldsymbol{\delta}, \theta\big) \approx -\frac{1}{2}\log(\theta) - \frac{1}{2}\log\Big|K_i''(\tilde{b}_i)\Big| - K_i(\tilde{b}_i) - c\boldsymbol{\gamma}(\boldsymbol{\omega}^T\boldsymbol{\omega} + \boldsymbol{\delta}^T\boldsymbol{\delta}),$$

where

$$K_i(\tilde{b}_i) = -\sum_{j=1}^{c}\Bigg[\Delta_{ij}\Big[\log\big(\lambda_0(t)\big) + \alpha_{ij}^{(L)} + \tilde{b}_i\Big] + \Lambda_0(t)\exp(\alpha_{ij}^{(L)} + \tilde{b}_i) + \frac{1}{2}\tilde{b}_i'\theta^{-1}\tilde{b}_i\Bigg]$$

and

$$K_i''(\tilde{b}_i) = \frac{\partial^2 K(\tilde{b}_i)}{\partial^2 \tilde{b}_i} = \sum_{j=1}^{c}\Big[\Lambda_0(t)\exp(\alpha_{ij}^{(L)} + \tilde{b}_i) + \theta^{-1}\Big].$$

The parameter $\tilde{b}_i = \tilde{b}_i(\boldsymbol{\omega}, \boldsymbol{\delta})$ denotes the solution to the partial derivatives of $K_i(b_i)$ with respect to b_i. According to Ye, Lin, and Taylor (2008), omitting the complicated term $\log|K_i''(\tilde{b}_i)|$ in $\log(\tilde{L}_i^{NN})$ has a negligible effect on the parameter estimation, so we remove it in the likelihood approximation. Further, for right-censored data, to avoid estimating the baseline hazard function, replacing the full likelihood in $K_i(\tilde{b}_i)$ with a partial likelihood leads to the following penalized approximated partial log-likelihood

$$pl = \sum_{i=1}^{s} pl_i = \sum_{i=1}^{s}\sum_{j=1}^{c}\Bigg\{\Delta_{ij}\Bigg[(\alpha_{ij}^{(L)} + b_i) - \log\sum_{d,q\in R(t_{ij})}\exp(\alpha_{dq}^{(L)} + b_q)\Bigg] - \frac{1}{2}b_i'\theta^{-1}b_i\Bigg\} - n\boldsymbol{\gamma}(\boldsymbol{\omega}^T\boldsymbol{\omega} + \boldsymbol{\delta}^T\boldsymbol{\delta}), \quad (3)$$

where $R(t_{ij})$ denotes indexes for subjects who are at risk at time t_{ij}.

With the partial likelihood, given θ, intuitively we can estimate $(\boldsymbol{\omega}, \boldsymbol{\delta})$ by solving $\frac{\partial pl}{\partial \boldsymbol{\omega}} = 0$, and $\frac{\partial pl}{\partial \boldsymbol{\delta}} = 0$. Then the random effect can be updated by solving $\frac{\partial pl}{\partial b_i} = 0$ given the updated $(\boldsymbol{\omega}, \boldsymbol{\delta})$ from the first step. The two steps are iterated until convergence. However, to avoid the computation burden caused by this iterative algorithm, we proposed the following loss function for the estimation of $(\boldsymbol{\omega}, \boldsymbol{\delta})$ and random effect b_i instead, so the parameters $\{\boldsymbol{\omega}, \boldsymbol{\delta}, \eta_{ij}^{(x)}, \hat{b}_i = \hat{\eta}_i^{(b)}\}$ can be updated together in neural network back-propagation,

$$plnn = \sum_{i=1}^{s}\sum_{j=1}^{c}\Bigg\{\Delta_{ij}\Bigg[(\eta_{ij}^{(x)}\alpha_{ij}^{(L)} + \eta_i^{(b)}) - \log\sum_{d,q\in R(t_{ij})}\exp(\eta_{dq}^{(x)}\alpha_{dq}^{(L)} + \eta_q^{(b)})\Bigg] - \frac{1}{2}\eta_i^{(b)\prime}\theta^{-1}\eta_i^{(b)}\Bigg\} - n\boldsymbol{\gamma}(\boldsymbol{\omega}^T\boldsymbol{\omega} + \boldsymbol{\delta}^T\boldsymbol{\delta}), \quad (4)$$

where $\alpha_{ij}^{(L)} = g_L(\boldsymbol{\omega}^{(L)}\boldsymbol{\alpha}_{ij}^{(L-1)})$, $\boldsymbol{\eta}^{(x)} = (\eta_{11}^{(x)}, ..., \eta_{sc}^{(x)})$ and $\boldsymbol{\eta}^{(b)} = (\eta_1^{(b)}, ...\eta_c^{(b)})$ are weights for the final output layer. Please refer to Figure 1 as a clear demonstration of our proposed neural network structure. With this structure, instead of estimating neural network parameters and random effect b_i iteratively, we can estimate and update the random effect b_i in one step using $\hat{\eta_i}^{(b)}$, and $\hat{\boldsymbol{\eta}}^{(b)}$ is updated together with other parameters $(\hat{\boldsymbol{\omega}} = \{\hat{\boldsymbol{\omega}}^{(0)}, ..., \hat{\boldsymbol{\omega}}^{(L)}\}, \hat{\boldsymbol{\delta}} = \{\hat{\boldsymbol{\delta}}^{(0)}, ..., \hat{\boldsymbol{\delta}}^{(L-1)}\}, \hat{\boldsymbol{\eta}}^{(x)})$. For a single-layer network, differentiation of the approximated partial likelihood with respect to $\boldsymbol{\eta}^{(x)}$, $\boldsymbol{\eta}^{(b)}$, $\boldsymbol{\omega}$, $\boldsymbol{\delta}$ leads to the following quasi-score equations with $\alpha_{ij}^{(1)} = g_1(\boldsymbol{\omega}^{(1)}\boldsymbol{\alpha}_{ij}^{(0)})$:

$$\frac{\partial plnn}{\partial \eta_{ij}^{(x)}} = \Delta_{ij}\Bigg(\alpha_{ij}^{(1)} - \frac{\alpha_{ij}^{(1)}\exp(\eta_{ij}^{(x)}\alpha_{ij}^{(1)} + \eta_i^{(b)})}{\sum_{d,q\in R(t_{ij})}\exp(\eta_{dq}^{(x)}\alpha_{dq}^{(1)} + \eta_q^{(b)})}\Bigg), \quad (5)$$

$$\frac{\partial plnn}{\partial \eta_i^{(b)}} = \sum_{j=1}^{c}\Delta_{ij}\Bigg(1 - \frac{\exp(\eta_{ij}^{(x)}\alpha_{ij}^{(1)} + \eta_i^{(b)})}{\sum_{d,q\in R(t_{ij})}\exp(\eta_{dq}^{(x)}\alpha_{dq}^{(1)} + \eta_q^{(b)})}\Bigg), \quad (6)$$

$$\frac{\partial plnn}{\partial \omega_{k1}^{(1)}} = \sum_{i=1}^{s}\sum_{j=1}^{c}\Delta_{ij}\Bigg(\eta_{ij}^{(x)} -$$

$$\left(\frac{\eta_{ij}^{(x)}\exp\left(\eta_{ij}^{(x)}\alpha_{ij}^{(1)}+\eta_{i}^{(b)}\right)}{\sum_{d,q\in R(t_{ij})}\exp\left(\eta_{dq}^{(x)}\alpha_{dq}^{(1)}+\eta_{q}^{(b)}\right)}\right)\cdot g_1'(\boldsymbol{\omega}^{(1)}\boldsymbol{\alpha}_{ij}^{(0)})$$
$$g_0(\omega_{.k}^{(0)}x_{ij}+\delta_k^{(0)})-2n\gamma\omega_{k1}^{(1)}, \quad (7)$$

$$\frac{\partial plnn}{\partial\omega_{lk}^{(0)}}=\sum_{i=1}^{s}\sum_{j=1}^{c}\Delta_{ij}\left(\eta_{ij}^{(x)}-\right.$$
$$\left.\frac{\eta_{ij}^{(x)}\exp\left(\eta_{ij}^{(x)}\alpha_{ij}^{(1)}+\eta_{i}^{(b)}\right)}{\sum_{d,q\in R(t_{ij})}\exp\left(\eta_{dq}^{(x)}\alpha_{dq}^{(1)}+\eta_{q}^{(b)}\right)}\right)g_1'(\boldsymbol{\omega}^{(1)}\boldsymbol{\alpha}_{ij}^{(0)})$$
$$\omega_{k1}^{(1)}g_0'(\omega_{.k}^{(0)}x_{ij}+\delta_k^{(0)})x_{ijl}-2n\gamma\omega_{lk}^{(0)} \quad (8)$$

$$\frac{\partial plnn}{\partial\delta_k^{(0)}}=\sum_{i=1}^{s}\sum_{j=1}^{c}\Delta_{ij}\left(\eta_{ij}^{(x)}-\right.$$
$$\left.\frac{\eta_{ij}^{(x)}\exp\left(\eta_{ij}^{(x)}\alpha_{ij}^{(1)}+\eta_{ij}^{(b)}\right)}{\sum_{d,q\in R(t_{ij})}\exp\left(\eta_{dq}^{(x)}\alpha_{dq}^{(1)}+\eta_{q}^{(b)}\right)}\right).$$
$$g_1'(\boldsymbol{\omega}^{(1)}\boldsymbol{\alpha}_{ij}^{(0)})\omega_{k1}^{(1)}g_0'(\omega_{.k}^{(0)}x_{ij}+\delta_k^{(0)})-2n\gamma\delta_k^{(0)} \quad (9)$$

where $\boldsymbol{\eta}^{(x)}$ and $\boldsymbol{\eta}^{(b)}$ are the weights for the last layer, $\omega_{k1}^{(1)}$ is the weight connecting the kth hidden node to the univariate output $\alpha_{ij}^{(1)}$, $\omega_{lk}^{(0)}$ is the weight connecting the lth input to the kth hidden node in hidden layer, $\delta_k^{(0)}$ is the bias of the kth hidden node in hidden layer, and $\omega_{.k}^{(0)}$ is the kth entry of the vector $\omega^{(0)}$.

To train the neural network, we develop our code along the lines of the Deepsurv method: standardization of the continuous input, Adaptive Moment Estimation (Adam) for the gradient descent algorithm, Nesterov momentum, and learning rate schedule. Adam for gradient descent algorithm (Kingma and Ba (2015)) is an algorithm for first-order gradient-based optimization of the objective function, it accelerated the gradient descent algorithm by taking into consideration the "exponentially weighted average" of the gradients. Thus the algorithm converges towards the minima at a faster pace. Nesterov momentum helps to avoid first-order optimization problems, e.g., exploding gradients. We apply the exponential learning rate decay constant and inverse time decay to the learning rate at each epoch. Since the goal is on prediction, we will focus on the estimation of $(\boldsymbol{\omega}, \boldsymbol{\delta}, \boldsymbol{\eta}^{(x)}, \boldsymbol{\eta}^{(b)})$. The parameter θ is estimated by solving the estimating equation derived from penalized partial likelihood function as in Ripatti and Palmgren (2000). The baseline hazard function can be estimated with a Breslow-type estimator, with

$$\widehat{\Lambda}_0(t)=\sum_{i,j:x_{ij}\leq \mathrm{t}}\frac{\Delta_{ij}}{\Sigma_{d,q\in R(x_{ij})}\exp\left(\hat{\eta}_{dq}^{(x)}g_1(\hat{\boldsymbol{\omega}}^{(1)}\hat{\boldsymbol{\alpha}}_{dq}^{(0)})+\hat{\eta}_q^{(b)}\right)}$$

3. Simulation Study

We generate the data under the Cox model with shared frailty and nonlinear effects (true model). Then we compare the proposed method to (i) Deepsurv, (ii) the Cox model with only linear effects, (ii) the Cox model with linear effects and interactions, (iv) the Cox model with frailty and linear effects, and (v) the Cox model with frailty, linear and interaction effects.

To compare the prediction accuracy of all those methods, we measure the concordance-index (C-index), the most frequently used evaluation metric in survival analysis (Harrell et al. 1984). It is a measure of the rank correlation between predicted risk scores and observed time points. If C-index$= 0.5$, the prediction method is the same as a random guess. If C-index$= 1$, the ranking of predicted risk scores perfectly matches that of the observed death times. As in the real data analysis, we are interested in the within-cluster prediction; so for pairs within a cluster, we randomly assign one subject to the training dataset and the other one to the test dataset. The Relu activation function is selected in the neural network prediction.

The data are generated from a proportional hazards model,

$$\lambda_{ij}\left(t \mid b_i\right)=\lambda_0(t)\exp(\boldsymbol{X}_{ij}^T\boldsymbol{\beta}+b_i),$$

where $i=1,\ldots,s$, $j=1,\ldots,c$, with $\Lambda_0(t)=t$. To mimic the observations in the motivating example OHTS dataset, we have $s=1500$ clusters and cluster size $c=2$. Four values for the frailty variance are used, i.e., $\theta=0, 1.5, 2.5,$ and 3.5. We consider two scenarios for $\boldsymbol{X}_{ij}=(X_{ij1},X_{ij2},X_{ij3},X_{ij4},X_{ij5})^T$. In both scenarios, we first generate $\boldsymbol{M}_{ij}=(M_{ij1},M_{ij2},M_{ij3},M_{ij4},M_{ij5})^T$ from a multivariate normal distribution with mean and covariance matrix calculated from the five baseline covariates age, intraocular pressure (IOP), central corneal thickness [CCT], pattern standard deviation [PSD], and vertical cup disc ratio [VCD] in the OHTS dataset. Then $\boldsymbol{M}_{ij}$ are standardized to generate nonlinear effects $\boldsymbol{X}_{ij}$. In scenario 1, following the setup in Katzman et al. (2018), the covariates are calculated by $\boldsymbol{X}_{ij}=\left(M_{ij1}^2,M_{ij2}^2,M_{ij3}^2,M_{ij4}^2,M_{ij5}^2\right)^T$, and the parameters are $\boldsymbol{\beta}=(\beta_1,\beta_2,\beta_3,\beta_4,\beta_5)^T=(0.25,0.25,0.25,0.25,0.25)^T$. The censoring times are generated from $Uniform(0,3)$ with around 20% of the event times are independently right censored, and $Uniform(0,1)$ with around 40% of the event times are independently right censored. In scenario 2, we generate more complicated nonlinear effects following case 3 in Zhong et al. (2022): $\boldsymbol{X}_{ij}=\{\boldsymbol{X}_{ij1},\boldsymbol{X}_{ij2},\boldsymbol{X}_{ij3},\boldsymbol{X}_{ij4}\}=\{M_{ij1}^2M_{ij2}^3,\log\left(|M_{ij3}|+1\right),\sqrt{|M_{ij4}M_{ij5}|+1},\exp\left(\frac{M_{ij5}}{2}\right)\}$. The censoring times are generated from $Uniform(0,4.5)$ with around 20% of right censoring rate and from $Uniform(0,1.5)$ with around 40% of right censoring rate.

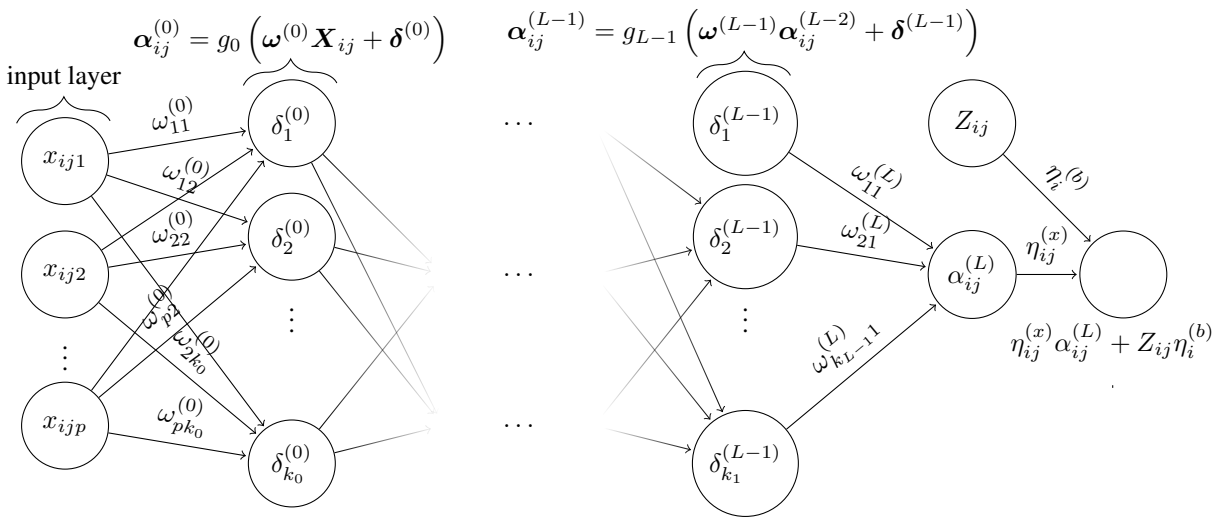

Figure 1. Network graph of a $(L+1)$-layer perceptron with p input units. The random effect b_i and covariates $Z_{ij} = 1$ are included in the final layer of the network.

Tables 1 and 2 show the results under different frailty variances with 20% and 40% censoring rates under scenario 1, respectively. Tables 3 and 4 show the results with 20% and 40% censoring rates under scenario 2, respectively. The proposed method and Deepsurv are fitted under a two-layer neural network with $(32, 32)$ and $(64, 64)$ numbers of hidden nodes. When random effect variance $\theta = 0$, which corresponds to no correlation between subjects within a cluster, the C-indexes of the proposed method are smaller than Deepsurv. As θ increases, the proposed method performs consistently better than the five competing methods, and the C-indexes of the proposed method are closer to the true model. Compared to Deepsurv, the proposed method includes frailty terms for cluster effect. Thus, the disparity between the proposed method and Deepsurv becomes more significant as θ increases. In general, when no correlation between subjects within a cluster exists, Deepsurv performs better than the four Cox proportional hazards models, which demonstrates the significant improvement of using a feed-forward neural network when the log-hazard or risk scores are nonlinear.

4. OHTS Data Analysis

We compare the accuracy of the proposed method with five other competing methods in predicting time to the development of primary open-angle glaucoma (POAG). We use data from a randomized clinical trial of the Ocular Hypertension Treatment Study (OHTS). The OHTS is designed to test the safety and efficacy of topical ocular hypotensive medication in delaying or preventing the development of POAG in individuals with ocular hypertension. The analysis cohort contains 1636 participants with ocular hypertension who were randomized to receive either topical ocular hypotensive medication (medication group) or close observation (observation group). Following Gordon et al. (2020), we include five baseline factors (age, intraocular pressure [IOP], central corneal thickness [CCT], pattern standard deviation [PSD], and vertical cup disc ratio [VCD]). In our analysis, the event time T_{ij} is defined as the duration from baseline to the date of the first abnormal visual field or optic disc photograph that masked readers determined met the criteria for reproducible change and the endpoint committee attributed to POAG. Among the 1636 patients,483 developed glaucoma by the end of the study, of whom 226 subjects developed glaucoma in both eyes. We excluded 190 subjects with missing baseline covariate measurements. Thus, the sample size used in this analysis is 1446 with 2892 eyes. The follow-up is censored after drop-out of the study, end of the study, or death, with a censoring rate of 76%.

As in Simulation study, we compare our model to the five competitive models to predict the time to POAG with five baseline factors age, IOP, CCT, PSD, and VCD. Each subject is regarded as a cluster and each cluster contains two eyes. We use one eye from each patient as the training dataset and the other eye as the testing dataset. The goal of our analysis is to use the time-to-glaucoma of one eye to predict that of the other eye on the same patient. Accurate within-subject prediction is important to identify patients who are more likely to have a visual loss to the other eye to prevent irreversible damage or unnecessary over-treatment. The Relu activation function is used for prediction. Table 5 reports the C-indexes by using one eye to predict the time to POAG of the other eye using six methods. The proposed method has the highest C-index among all the methods, which indicates the non-linearity and clustering effect in risk function. By considering the clustering and non-linearity co-

Table 1. C-index on the 100 simulated test data sets on scenario 1 under 20% censoring rate.

Frailty variance	Hidden Nodes	Proposed	Deepsurv	Cox frailty(Linear)	Cox frailty with interactions	Cox	Cox with interactions	True Model
0	(32, 32)	63.12	65.77	49.93	55.22	49.93	55.13	69.14
0	(64, 64)	62.95	65.02	49.93	55.22	49.93	55.13	69.14
1.5	(32, 32)	60.22	59.54	49.77	53.48	49.76	53.52	64.13
1.5	(64, 64)	61.03	59.20	49.77	53.48	49.76	53.52	64.13
2.5	(32, 32)	60.24	57.64	49.80	52.91	49.80	52.96	62.42
2.5	(64, 64)	60.47	57.54	49.80	52.91	49.80	52.96	62.42
3.5	(32, 32)	60.65	56.51	49.83	52.32	49.83	52.39	60.57
3.5	(64, 64)	60.65	56.33	49.83	52.32	49.83	52.39	60.57

Table 2. C-index on the 100 simulated test data sets on scenario 1 under 40% censoring rate.

Frailty variance	Hidden Nodes	Proposed	Deepsurv	Cox frailty(Linear)	Cox frailty with interactions	Cox	Cox with interactions	True Model
0	(32, 32)	63.95	65.77	49.92	54.88	49.92	54.88	68.37
0	(64, 64)	61.15	65.34	49.92	54.88	49.92	54.88	68.37
1.5	(32, 32)	61.23	59.88	49.78	53.19	49.80	53.16	63.33
1.5	(64, 64)	61.39	59.39	49.78	53.19	49.80	53.16	63.33
2.5	(32, 32)	60.91	58.03	49.80	52.69	49.80	52.66	61.71
2.5	(64, 64)	61.07	57.03	49.80	52.69	49.80	52.66	61.71
3.5	(32, 32)	60.85	56.51	49.80	52.51	49.81	52.57	61.20
3.5	(64, 64)	60.78	56.33	49.80	52.51	49.81	52.57	61.20

Table 3. C-index on the 100 simulated test data sets on scenario 2 under 20% censoring rate.

Frailty variance	Hidden Nodes	Proposed	Deepsurv	Cox frailty(Linear)	Cox frailty with interactions	Cox	Cox with interactions	True Model
0	(32, 32)	55.63	59.94	58.93	58.69	58.92	58.69	62.19
0	(64, 64)	57.96	59.12	58.93	58.69	58.92	58.69	62.19
1.5	(32, 32)	57.94	56.30	57.24	56.82	57.23	56.82	63.11
1.5	(64, 64)	58.48	56.50	57.24	56.82	57.23	56.82	63.11
2.5	(32, 32)	59.25	56.38	56.55	56.16	56.55	56.17	64.18
2.5	(64, 64)	58.76	55.66	56.55	56.16	56.55	56.17	64.18
3.5	(32, 32)	59.71	54.91	56.06	55.61	56.07	55.61	64.55
3.5	(64, 64)	59.38	54.83	56.06	55.61	56.07	55.61	64.55

Table 4. C-index on the 100 simulated test data sets on scenario 2 under 40% censoring rate.

Frailty variance	Hidden Nodes	Proposed	Deepsurv	Cox frailty(Linear)	Cox frailty with interactions	Cox	Cox with interactions	True Model
0	(32, 32)	56.80	59.44	59.18	58.87	59.17	58.85	62.46
0	(64, 64)	56.40	59.03	59.18	58.87	59.17	58.85	62.46
1.5	(32, 32)	58.28	56.60	57.54	57.06	57.52	57.03	63.17
1.5	(64, 64)	58.00	56.46	57.54	57.06	57.52	57.03	63.17
2.5	(32, 32)	58.49	55.54	56.73	56.28	56.72	56.24	64.23
2.5	(64, 64)	58.75	55.41	56.73	56.28	56.72	56.24	64.23
3.5	(32, 32)	59.20	55.20	56.32	55.86	56.31	55.82	65.10
3.5	(64, 64)	59.94	54.82	56.32	55.86	56.31	55.82	65.10

Table 5. C-index on the OHTS data set

Proposed	Deepsurv	Cox frailty	Cox frailty with interactions	Cox	Cox with interactions
71.69	67.28	69.89	69.92	69.83	69.82

variate effects, the proposed method improves the prediction accuracy.

5. Conclusion

We propose a neural network for correlated survival outcomes. The proposed method extends the classical feed-forward neural network framework to include a random effect (frailty) accounting for within-cluster correlation. The model uses a feed-forward neural network for nonlinear fixed effects and estimates random effects in the last layer of the neural network to avoid iterative computation. The neural network is trained over a loss function derived from the penalized partial likelihood with a Laplace approximation. We demonstrate the benefits of our approach compared to other traditional survival regression methods and Deepsurv in simulation studies and real data analysis. The proposed method is a powerful tool for predicting correlated survival outcomes in the presence of complicated covariate effects.

There are several directions for future research. First, in this paper we only consider correlated survival outcomes in clusters. It is of interest to develop a method for recurrent event data - another form of the correlated survival outcomes (Cook and Lawless 2007). Second, it would be of interest to develop deep learning prediction methods for correlated survival outcomes using e.g., the additive hazards model (Aalen 1989) or the linear transformation model (Fine et

al. 1998). Finally, we only consider time-independent covariates (at baseline) for prediction of time to event. It is of importance to consider longitudinal biomarkers for dynamic prediction in the joint model and landmark model frameworks (Rizopoulos et al. 2017, Tanner et al. 2021, Lin and Luo 2022).

Reference

Aalen, O. O. (1989). A linear regression model for the analysis of life times. *Statistics in Medicine*, 8, 907-925.

Balan, T. A., and Putter, H. (2020). A tutorial on frailty models. *Statistical Methods in Medical Research*, 29(11), 3424-3454.

Ching, T., Zhu, X., and Garmire, L. X. (2018). Cox-nnet: an artificial neural network method for prognosis prediction of high-throughput omics data. *PLoS Computational Biology*, 14(4), e1006076.

Cook, R. J., Lawless, J. F. (2007). *The statistical analysis of recurrent events*. Springer Science & Business Media.

Faraggi, D., and Simon, R. (1995). A neural network model for survival data. *Statistics in Medicine*, 14(1), 73-82.

Fine, J. P., Ying, Z., Wei, L. J. (1998). On the Linear Transformation Model for Censored Data. *Biometrika*, 85, 980-986.

Gordon, M. O., Gao, F., Huecker, J. B., et al. (2020). Evaluation of a primary open-angle glaucoma prediction model using long-term intraocular pressure variability data: a secondary analysis of 2 randomized clinical trials. *JAMA Ophthalmology*, 138(7), 780-788.

Glidden, D. V., and Vittinghoff, E. (2004). Modelling clustered survival data from multicentre clinical trials. *Statistics in Medicine*, 23(3), 369-388.

Harrell, F. E., Lee, K. L., Califf, R. M., Pryor, D. B., and Rosati, R. A. (1984). Regression modelling strategies for improved prognostic prediction. *Statistics in Medicine*, 3(2), 143-152.

Hao, J., Kim, Y., Mallavarapu, T., Oh, J. H., and Kang, M. (2018, December). Cox-PASNet: pathway-based sparse deep neural network for survival analysis. *In 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (pp. 381-386). IEEE.

Hens, N., Wienke, A., Aerts, M., and Molenberghs, G. (2009).The correlated and shared gamma frailty model for bivariate current status data: an illustration for cross-sectional serological data. *Statistics in Medicine*, 28(22), 2785-2800.

Jeanselme, V., Tom, B., and Barrett, J. (2022, April). Neural Survival Clustering: Non-parametric mixture of neural networks for survival clustering. *Conference on Health, Inference, and Learning* (pp. 92-102). PMLR.

Kingma, D. P., and Ba, J. (2015). Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR) 2015*

Kvamme H, Borgan Nø, and Scheel I.(2019). Time-to-event prediction with neural networks and Cox regression. *Journal of Machine Learning Research* 20, 1-30

Kass, M. A., Heuer, D. K., Higginbotham, E. J., Johnson, C. A., Keltner, J. L., Miller, J. P., ... and Ocular Hypertension Treatment Study Group. (2002). The Ocular Hypertension Treatment Study: a randomized trial determines that topical ocular hypotensive medication delays or prevents the onset of primary open-angle glaucoma. *Archives of Ophthalmology*, 120(6), 701-713.

Kass, M. A., Heuer, D. K., Higginbotham, E. J., Parrish, R. K., Khanna, C. L., Brandt, J. D., ... and Ocular Hypertension Study Group. (2021). Assessment of cumulative incidence and severity of primary open-angle glaucoma among participants in the ocular hypertension treatment study after 20 years of follow-up. *JAMA Ophthalmology*, 139(5), 558-566.

Kass, M. A., Gordon, M. O., Gao, F., Heuer, D. K., Higginbotham, E. J., Johnson, C. A., ... and Wilson, M. R. (2010). Delaying treatment of ocular hypertension: the ocular hypertension treatment study. *Archives of Ophthalmology*, 128(3), 276.

Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T., and Kluger, Y. (2018). DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC Medical Research Methodology*, 18(1), 1-12.

Liao, L., and Ahn, H. I. (2016). Combining deep learning and survival analysis for asset health management. *International Journal of Prognostics and Health Management*, 7(4).

Lin, J., and Luo, S. (2022). Deep learning for the dynamic prediction of multivariate longitudinal and survival data. *Statistics in Medicine.* 41(15), 2894-2907.

Martinsson, E. (2016). Wtte-rnn: Weibull time to event recurrent neural network (Doctoral dissertation, Chalmers University of Technology and University of Gothenburg).

Mandel, F., Ghosh, R. P., and Barnett, I. (2021). Neural networks for clustered and longitudinal data using mixed effects models. *Biometrics*.

Manduchi, L., Marcinkevičs, R., Massi, M. C., Weikert, T., Sauter, A., Gotta, V., ... and Vogt, J. E. (2021). A deep variational approach to clustering survival data. *arXiv preprint arXiv*:2106.05763.

Paik, M. C., Tsai, W. Y., and Ottman, R. (1994). Multivariate survival analysis using piecewise gamma frailty. *Biometrics*, 975-988.

Ripatti, S., and Palmgren, J. (2000). Estimation of multivariate frailty models using penalized partial likelihood. *Biometrics*, 56(4), 1016-1022.

Ranganath, R., Perotte, A., Elhadad, N., and Blei, D. (2016, December). Deep survival analysis. *In Machine Learning for Healthcare Conference* (pp. 101-114). PMLR.

Rizopoulos, D., Molenberghs, G., and Lesaffre, E. M. (2017). Dynamic predictions with time-dependent covariates in survival analysis using joint modeling and landmarking. *Biometrical Journal*, 59(6), 1261-1276.

Shih, J. H., and Louis, T. A. (1995). Assessing gamma frailty models for clustered failure time data. *Lifetime Data Analysis*, 1(2), 205-220.

Tanner, K. T., Sharples, L. D., Daniel, R. M., and Keogh, R. H. (2021). Dynamic survival prediction combining landmarking with a machine learning ensemble: Methodology and empirical comparison. *Journal of the Royal Statistical Society: Series A (Statistics in Society),* 184(1), 3-30.

Ye, W., Lin, X., and Taylor, J. M. (2008). Semiparametric modeling of longitudinal measurements and time-to-event data-a two-stage regression calibration approach *Biometrics*, 64(4), 1238-1246.

Zhong, Q., Mueller, J., and Wang, J. L. (2022). Deep learning for the partially linear cox model. *The Annals of Statistics*, 50(3), 1348-1375.

Acknowledgements