Survival Prediction Using Deep Learning

Aliasghar Tarkhan ATARKHAN@UW.EDU
Noah Simon NRSIMON@UW.EDU
Department of Biostatistics, University of Washington, Seattle, WA
Thomas Bengtsson THOMASGB@GENE.COM
Kien Nguyen NGUYENK8@GENE.COM
Jian Dai DAIJ12@GENE.COM
PHC Imaging Group, Genentech, South San Francisco, CA

Abstract

In many biomedical applications, outcome is measured as a "time-to-event" (e.g., time-to-disease progression or death). Cox proportional hazards (CoxPH) model has been widely used to assess the association between baseline characteristics of a patient and this outcome. Meanwhile, in therapeutic areas such as Oncology, clinical imaging (e.g. computerized tomography (CT) scan) is widely used for detection, diagnosis of disease, monitoring of progression and treatment effect. We are interested in using such images with neural network to build predictive models with survival data. However, the standard methodologies cannot be applied to imaging data with time-to-event outcome due to challenges such as memory constraint. In this work, we develop a simple methodology to engage images with survival data. Our proposed methodology is a modified version of CoxPH model that is amenable to SGD and allows us to overcome the existing challenges. We present the neural network architecture for the survival prediction using images. Our architecture can leverage new advances in network topology.

Keywords: Survival prediction; Time-to-event outcome; Deep neural network; Convolutional neural network; Cox proportional hazards; Stochastic gradient descent; Imaging data.

1. Introduction

In many biomedical applications, outcome is measured as a time to an event of interest, e.g., time to death, time to disease progression, etc. Assessing the relationship between the baseline features of a patient and this outcome is known as Survival Analysis (Schober and Vetter, 2018). In such applications, we have partial information on some patients due to censoring. For instance, some patients may leave the study early before experiencing the event of interest. The Cox proportional hazards (CoxPH) model (Cox, 1972) is a widely used tool for assessing such association when data are incomplete due to censoring. In such a model with n patients, the parameter of interest $\boldsymbol{\beta}$ is estimated by maximizing an objective function called the *log-partial-likelihood* which is defined as

$$pl^{(n)}(\boldsymbol{\beta}|\mathcal{D}^{(n)}) = \sum_{i=1}^{n} \delta_i \left(f_{\boldsymbol{\beta}}(\boldsymbol{x}^{(i)}) - \log \Big(\sum_{j \in \mathcal{R}_i} \exp\left(f_{\boldsymbol{\beta}}(\boldsymbol{x}^{(j)})\right) \Big) \right) \tag{1}$$

where covariate $\boldsymbol{x}^{(i)}$ is a p-dimensional vector representing characteristics of patient i; $f_{\boldsymbol{\beta}}(\boldsymbol{x}^{(i)})$ is a specified function, usually called the risk function. It connects the covariates (characteristics) of interest $\boldsymbol{x}^{(i)}$ to the outcome of interest (survival time). In many

scenarios where the standard CoxPH model is used, f is simply chosen as a linear function, i.e., $f_{\boldsymbol{\beta}}(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{\beta} = +\beta_1 x_1 + \cdots + \beta_p x_p$; $\mathcal{R}_i = \{j \,|\, t_j \geq t_i\}$ is the "risk set for patient i"; δ_i is an indicator showing whether patient i is censored ($\delta_i = 1$: not censored, otherwise censored). $\mathcal{D}^{(n)}$ represents n independent observations drawn from Cox proportional hazards model: $\mathcal{D}^{(n)} = \{\mathcal{D}_i = (y_i, \delta_i, \boldsymbol{x}^{(i)}) | i = 1, 2, \ldots, n\}$ where y_i is time to event or censoring, whichever comes first.

Although the standard CoxPH model has been widely used, it is not amenable to stochastic gradient descent (SGD)-based algorithms because the expression in (1) cannot be split over individual patients. SGD-based algorithms are key to engage with complex prediction models such as those characterized by neural networks. In applications such as oncology and pathology where we have imaging data (e.g., CT or pathological images), we train complex models using SGD-based algorithms to do prediction/classification. Therefore, there is a need to come up with a method that facilitates using SGD-based algorithm for survival prediction (i.e., with time-to-event outcome) through training complex models for $f_{\boldsymbol{\beta}}(\boldsymbol{x})$ through neural networks.

There have been some efforts to overcome this issue with standard CoxPH models. Toulis and Airoldi (2017) presented SGD-based algorithms for a variety of applications including the Cox proportional hazards model. However their algorithm suffers from two issues: It cannot accommodate streaming data, and in fact it has high computational complexity (it requires $\sim n^2$ computation). Raykar et al. (2008) proposed directly maximizing the concordance index (Austin and Steyerberg, 2012). While this is an interesting predictive target, it moves us away from generative parameters in the Cox model. Katzman et al. (2017); Ching et al. (2018) connected neural networks to the log-partial likelihood. However, they engaged with [non-stochastic] gradient descent, which is not amenable to very large and/or imaging datasets. The work of Kvamme et al. (2019) is most closely related work. As with Katzman et al. (2017) and Ching et al. (2018), they connect neural networks with the partial likelihood; however they note that a stochastic gradient-like optimization method will be needed to scale to large datasets. As such, they come up with a heuristic for an "approximate gradient". They do not justify the heuristic (and in fact, their stochastic gradient is not unbiased, so there is no guarantee that any of the results of SGD-based methods will hold). In (Gensheimer and Narasimhan, 2019) authors presented a *fully-parametric* model based on SGD that is out of our focus that is CoxPH as a semi-parametric model.

Authors in (Tarkhan and Simon, 2020) presented a simple framework for proportional hazards regression that is amenable to SGD-based algorithms. It facilitates training complex, e.g., neural-network-based models with survival data. Their proposed framework generalizes and justifies some of the heuristic algorithms in other work — it both identifies why it should work and proves that it will. They considered a population parameter $\boldsymbol{\beta}^{(s)}$ that is defined as the population minimizer of the expected negative partial likelihood of s random patients as

$$\boldsymbol{\beta}^{(s)} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \Big\{ \mathbb{E}_s[-pl^{(s)}(\boldsymbol{\beta}|\mathcal{D}^{(s)})] \Big\} \tag{2}$$

where $\mathcal{D}^{(s)}$ is a draw of s random patients from the population. They proved that when the assumptions of the Cox model hold, then $\boldsymbol{\beta}^{(s)}$ is equal to the true parameter $\boldsymbol{\beta}^*$. Then

they proposed to use SGD where the estimated parameter at m-th iteration is given as

$$\hat{\boldsymbol{\beta}}(m) = \hat{\boldsymbol{\beta}}(m-1) + \gamma_m \times \nabla_\beta \left\{ pl^{(s)}(\hat{\boldsymbol{\beta}}(m-1)|\mathcal{D}_m^{(s)}) \right\}, \tag{3}$$

where $\hat{\boldsymbol{\beta}}(m-1)$ is the estimated parameter at the previous iteration $m-1$; γ_m is the learning rate; $\nabla_{\boldsymbol{\beta}}(.)$ is the gradient with respect to parameter $\boldsymbol{\beta}$; $pl^{(s)}(\hat{\boldsymbol{\beta}}(m-1)|\mathcal{D}_m^{(s)})$ is log-partial likelihood with random dataset of s patients $\mathcal{D}_m^{(s)}$.

In this extended abstract, we aim to use the proposed framework in (Tarkhan and Simon, 2020) to show how we can develop predictive survival model using imaging data by engaging with deep neural networks (DNN) to train complex models. Note that if we choose $s = 2$, the log-partial likelihood in (2) becomes a smoothed version of concordance index given by

$$pl^{(2)}(\boldsymbol{\beta}|\mathcal{D}^{(2)}) = log\Big(\frac{e^{f_{\boldsymbol{\beta}}(\boldsymbol{x}^{(1)})}}{e^{f_{\boldsymbol{\beta}}(\boldsymbol{x}^{(1)})} + e^{f_{\boldsymbol{\beta}}(\boldsymbol{x}^{(2)})}}\Big) 1(t_1 < t_2) + log\Big(\frac{e^{f_{\boldsymbol{\beta}}(\boldsymbol{x}^{(2)})}}{e^{f_{\boldsymbol{\beta}}(\boldsymbol{x}^{(1)})} + e^{f_{\boldsymbol{\beta}}(\boldsymbol{x}^{(2)})}}\Big) 1(t_2 < t_1). \tag{4}$$

Therefore, our loss function is defined as the negation of (4). Note that due to censoring, for instance in the case of $s = 2$, not all pairs are comparable. A pair is comparable if the patient with smaller time is not censored. We do not update parameters of model (defined by neural network) for those non-comparable pairs.

2. Method

2.1. Survival neural network architecture

One important aspect of the proposed framework (2) by (Tarkhan and Simon, 2020) was to facilitate the use of neural network-based models with time-to-event outcomes. Figure 1 illustrates the network architecture of such a model for survival prediction using imaging data. This architecture receives s images from a random stratum of s patients out of n patients each time. There are s parallel lines of network `Net` for the stratum of s images. The network `Net` is exactly the same for all lines (input images), i.e., they share the same parameters (coefficients) of interest $\boldsymbol{\beta}$ that need to be estimated. Output of these s lines are the estimated risk scores $f_{\boldsymbol{\beta}}(\boldsymbol{x}_i)$ for patients $i = 1, 2, \ldots, s$ patients. After gathering these s estimated risk scores, we estimate the parameters of network $\boldsymbol{\beta}$ by minimizing the negative log-partial likelihood (as the objective function) following (3). If $s = 2$, this architecture is similar to a *Siamese-like* network (Koch et al., 2015). Note that network `Net` can have any structure with one single output, it could be a customized network or a complex state-of-the-art network.

2.2. Data generation mechanism

We now engage with empirical experiments using the network architecture presented in Figure 1: We use MNIST dataset (Deng, 2012) to simulate time-to-event outcome. The MNIST dataset is a standard benchmark dataset that has been used for the image classification purposes. This dataset contains $n_{train} = 60,000$ and $n_{test} = 10,000$ greyscale

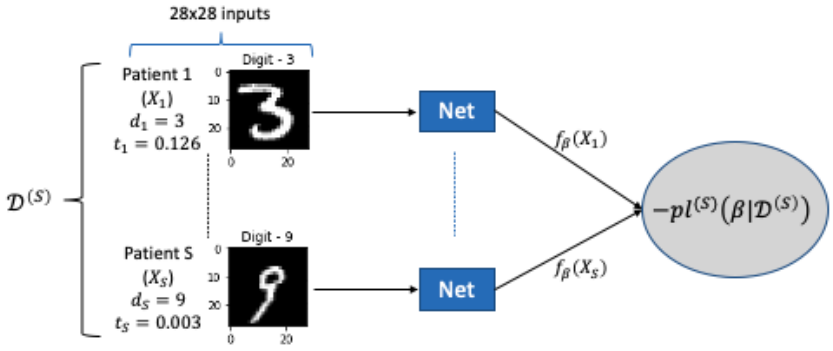

Figure 1: A generic network architecture for survival prediction using imaging data. There are s parallel lines of exactly the same network `Net`, one line for a single image. Output of each line is the estimated risk for corresponding patient which goes to the output node with a loss function defined in 2.

images with size 28x28 for training and testing, respectively. These images correspond to digit numbers between 0 and 9. We follow (Bender et al., 2005) to generate the censoring and event times independently assuming an exponential baseline hazards with parameter $\lambda = 1$. Time-to-event for digit d_i is given by

$$\begin{aligned} &y_i \sim exp(\mu = \exp(-\eta d_i)) \text{ (time to event/censoring)}, \\ &\delta_i \sim Bernoulli(p = 1 - p_c),\ p_c = Pr(t_i > c_i), \end{aligned} \tag{5}$$

where $y_i = min(t_i, c_i)$, i.e., time to event t_i or censoring c_i whichever comes first. Here p_c, the probability of censoring, is a parameter we can tune. In this work, we choose probability of censoring as 20% ($p_c = 0.2$); η is the proportion of risk score ($f_{\boldsymbol{\beta}}(\boldsymbol{X}_i)$) with respect to digit d_i. A higher η corresponds to a higher risk score and a better separation of times-to-event for different digits. Note that $\mu = \exp(-\eta d_i)$ is the scale parameter of exponential distribution. With datasets $\mathcal{D}^{(n_{train})} = (y_i, \delta_i, X_i), i = 1, 2, \ldots, n_{train}$ and assuming no information on d_i, the task of the neural network is to learn survival from handwritten images X_i.

2.3. Oracle concordance

For simulation results, we choose $s = 2$, i.e., we update our model using pairs of patients for each step of the SGD algorithm. For the sake of comparison, we calculate the Oracle concordance, i.e., the best concordance that the neural network can achieve as

$$C_{oracle}(\eta) = \frac{\sum_{i,j:t_i<t_j} \left\{ I[d_i > d_j] + 0.5 I[d_i = d_j] \right\}}{n_{test}(n_{test} - 1)/2}, \tag{6}$$

where $n_{test}(n_{test} - 1)/2$ represents the total number of possible pairs from testing dataset with testing data size n_{test}. In this definition, which has been widely used (Klaveren et al., 2016), we assign 1 if $t_i < t_j$ and $d_i > d_j$ (concordant), 0 if $t_i < t_j$ and $d_i < d_j$ (discordant), and 0.5 if $d_i = d_j$ (undecided). Note that the Oracle concordance depends on

η because distribution of time-to-event depends on η based on data generation mechanism in 5.

3. Results

We use strata size $s = 2$ with two lines of network `Net` where our objective function is defined as negative log of the smoothed concordance index defined in (4). Although, sub-network `Net` can have any arbitrary architecture, we consider two choices: (1) a minimalist architecture including three convolution layers and (2) a state-of-the-art network architecture EfficientNet (Howard et al., 2019) as part of network `Net`. In all of our simulations, we use AMSgrad algorithm with learning rate 10^{-4}, one stratum (pair) of patients per optimization step, and 100 epochs. For each epoch, we randomly split the training images into pairs of images.

3.1. Minimalist network

Given the fact that MNIST dataset includes small-sized greyscale images, a simplistic convolution neural network (CNN) has adequate receptive field to capture the relevant features. We choose a minimalist CNN network including three convolution layers with 32, 64, and 128 (5,5) filters with stride 1. Each of these three convolution layers are activated by ReLU functions and are followed by (2,2) Maxpool layers. Finally, all of these layers are followed by a fully connected layer with 256 nodes. Figure 2(a) compares testing concordance index for varying training sample sizes n and choices of η. As expected, the performance improves (closer to Oracle concordance) as η and/or n increases.

3.2. Transfer learning with state-of-the-art network

There are different ways to transfer the information learned by a pre-trained network from other problems to our specific problem. This is known as transfer learning Yang (2010). Depending on sample size and similarity of domain knowledge, we can freeze some layers (initial layers) and train others (higher layers) partially. In our simulation results, we use baseline EfficientNet-B0 as the first part of network `Net` to extract features with shape (7x7x1280) from the layer just before the softmax layer with 1000 outputs. After this feature extraction step, we added two dense layers with 500 and 256 nodes. Their parameters are updated by minimizing the objective function defined as negative log of concordance defined in (4). Figure 2(b) presents the concordance indices for varying n and η using EfficientNet-B0 as part of `Net`. The performance improves by increasing n and η. However, we observe that the performance is worse than that obtained using the minimalist network. The reason could be that the extracted features from EfficientNet-B0 are not strongly associated with the time-to-event outcome (due to difference in domain knowledge).

4. Discussion

We presented a generic neural network architecture for survival prediction using imaging data. We used the MNIST dataset to simulate time-to-event outcomes. We chose two networks for estimating the risk score: (1) a minimalist network and (2) EfficientNet as a

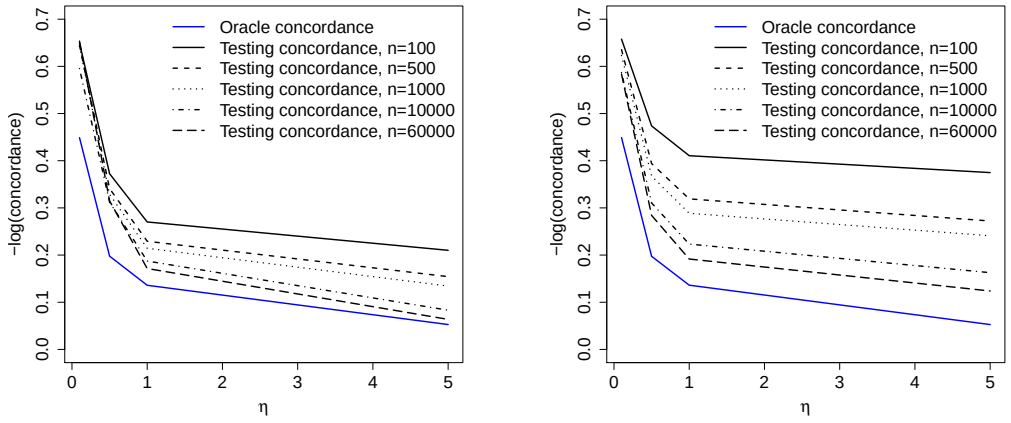

Figure 2: Negative log-concordance index for varying training sample sizes and choices of η (proportion of risk score to digit value) for (a) minimalist network including three convolution layers followed by a dense fully-connected layer with 256 nodes; (b) state-of-the-art network EfficientNet as a part of `Net` followed by two dense fully-connected layers with 500 and 256 nodes.

state-of-the-art network. We observed that both choices resulted in good performance in terms of concordance index.

It would be of interest to consider the performance of our algorithm with a variety of datasets. In this work we used the MNIST dataset as an illustrative example, as 1) it is straightforward to simulate event-times by defining risk scores proportional to digit numbers; and 2) MNIST is a standard dataset for which convolutional networks have shown strong performance for classification.

Although transfer learning with state-of-the-art networks may decrease computing time substantially, in our example the minimalist network for `Net` outperformed the state-of-the-art EfficientNet. This is likely, in part, because the MNIST dataset includes small-sized greyscale images of simple digits and a simplistic CNN likely suffices to capture the relevant features. However, for more complicated images, one may need to consider more complex (deeper and wider) networks and/or apply transfer learning with pre-trained state-of-the-art networks to get the best performance.

Acknowledgments

Thanks PHC Imaging Group at Genentech for the support.

References

P. C. Austin and E. W. Steyerberg. Interpreting the concordance statistic of a logistic regression model: relation to the variance and odds ratio of a continuous explanatory variable. *BMC Medical Research Methodology*, 12(82):1–8, 2012.

R. Bender, T. Augustin, and M. Blettner. Generating survival times to simulate cox proportional hazards models. *Statistics in medicine*, 24(11):1713–1723, 2005.

T. Ching, X. Zhu, and L. X. Garmire. Cox-nnet: An artificial neural network method for prognosis prediction of high-throughput omics data. *Plus Computational Biology*, 14(4): 1–18, 2018.

D. R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2):187–220, 1972.

L. Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

M. F. Gensheimer and B. Narasimhan. A scalable discrete-time survival model for neural networks. *PeerJ.*, 7, 2019.

A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Efficientnet: Rethinking model scaling for convolutional neural networks. eprint arXiv:1704.04861, 2019.

J. L. Katzman, U. Shaham, A. Cloninger, J. Bates, T. Jiang, and Y. Kluger. Deepsurv: Personalized treatment recommender system using a cox proportional hazards deep neural network. eprint arXiv:1606.00931v3, 2017.

D. V. Klaveren, M. Gonen, E. W. Steyerberg, and Y. Vergouwe. A new concordance measure for risk prediction models in external validation settings. *Statistics in medicine*, 35(23): 4136–4152, 2016.

G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. *Proceedings of the 32nd International Conference on Machine Learning*, 37, 2015.

H. Kvamme, O. Borgan, and I. Scheel. Time-to-event prediction with neural networks and cox regression. *Journal of Machine Learning Research*, 20(129):1–30, 2019.

V. C. Raykar, H. Steck, B. Krishnapuram, C. Dehing-Oberije, and P. Lambin. On ranking in survival analysis: Bounds on the concordance index. *Neural Information Processing Systems (NeurIPS)*, 2008.

P. Schober and T. R. Vetter. Survival analysis and interpretation of time-to-event data: The tortoise and the hare. *Anesthesia & Analgesia*, 127(3):792–798, 2018.

A. Tarkhan and N. Simon. Bigsurvsgd: Big survival data analysis via stochastic gradient descent. eprint arXiv:2003.00116, 2020.

P. Toulis and E. M. Airoldi. Asymptotic and finite-sample properties of estimators based on stochastic gradients. *The Annals of Statistics*, 45(4):1694–1727, 2017.

S. J. Pan; Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.